

# **SFA Critical Methods & Standards Approach**

## **SFA Modernization Program**

*US Department of Education*

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>1.1</b>	<b>Purpose</b>	<b>1</b>
<b>1.2</b>	<b>Scope</b>	<b>1</b>
<b>1.3</b>	<b>Current State of Methodology</b>	<b>2</b>
1.3.1	EDExpress/EDExpress Suite Development Methodology	2
1.3.2	TIV WAN Development Methodology	4
<b>1.4</b>	<b>Document Organization</b>	<b>5</b>
<b>1.5</b>	<b>Document Development Process Overview</b>	<b>6</b>
1.5.1	Documents Governing Methodology and Standards	6
1.5.2	Advisors	6
<b>2</b>	<b>Critical Methods</b>	<b>7</b>
<b>2.1</b>	<b>Capability Analysis Stage</b>	<b>7</b>
2.1.1	Collect and Manage Requirements	8
2.1.2	Define New or Modified Business Capability	10
2.1.3	Architect and Configure Release	15
2.1.4	Document Release for Vendor Bids	19
<b>2.2</b>	<b>Testing Process</b>	<b>21</b>
<b>3</b>	<b>Standards</b>	<b>22</b>
<b>3.1</b>	<b>Purpose</b>	<b>22</b>
<b>3.2</b>	<b>Stage Entry &amp; Exit Criteria</b>	<b>24</b>
3.2.1	Capability Analysis Stage	24
3.2.2	Capability Release Design Stage	25
3.2.3	Capability Release Build and Test Stage	27
3.2.4	Capability Deployment Stage	28
<b>4</b>	<b>Implementation Plan &amp; Compliance Monitoring</b>	<b>29</b>
<b>4.1</b>	<b>Implementation Plan</b>	<b>29</b>
<b>4.2</b>	<b>Compliance Monitoring Approach</b>	<b>30</b>
<b>5</b>	<b>APPENDIX</b>	<b>31</b>
<b>5.1</b>	<b>Quality Function Deployment</b>	<b>31</b>
5.1.1	The QFD Process	31
5.1.2	Performing QFD	34
5.1.3	QFD and Joint Application Development	35
<b>5.2</b>	<b>Business Performance Model Metrics and Definitions</b>	<b>36</b>
5.2.1	Sample Business Performance Model: Order and Delivery Management	36
<b>5.3</b>	<b>Business Capability Measurement and Communication Procedures</b>	<b>39</b>
5.3.1	Sample Business Capability: Measurement Procedures	39
5.3.2	Sample Business Capability: Performance Tracking	40

<b>5.4</b>	<b>Process Modeling</b>	<b>41</b>
5.4.1	Overview	41
5.4.2	Process Modeling Definitions	43
5.4.3	Process Modeling Standards and Guidelines	45
5.4.4	Swimlane Diagram Guidelines	57
<b>5.5</b>	<b>Sample Dialog Flow</b>	<b>64</b>
<b>5.6</b>	<b>Scenario Development Technique</b>	<b>66</b>
5.6.1	Workflow Scenarios	66
5.6.2	Scenario Gathering	68
5.6.3	Scenario Gathering Process	69
5.6.4	Tips and Traps	70
<b>5.7</b>	<b>Using a Variability Matrix</b>	<b>73</b>
5.7.1	Overview	73
5.7.2	Facilitation Process	73
5.7.3	Sample Workflow Variability Matrix	83
<b>5.8</b>	<b>Workflow Validation Guidelines</b>	<b>84</b>
<b>5.9</b>	<b>Deliverable Matrix</b>	<b>89</b>
<b>5.10</b>	<b>Entry and Exit Criteria</b>	<b>90</b>
5.10.1	Capability Analysis Stage	90
5.10.2	Capability Release Design Stage	91
5.10.3	Capability Release Build and Test Stage	93
5.10.4	Capability Deployment Stage	95

# **1 Introduction**

## **1.1 Purpose**

The primary purpose of the Critical Methods and Standards approach is to present a position on one of the two key areas of methodology that SFA follows, Requirements Management; the other key area is Testing. The secondary purpose is to set forth system development life cycle stage entry and exit standards for managing and controlling the quality of systems developed by contractors.

The multitude of projects and initiatives SFA is undertaking require not only an oversight program management function within the Modernization Partner, but it requires a strict set of guidelines and standards to hold the various development teams accountable. This model describes a methodology and standards program that the SFA could follow throughout the system development life cycle of the SFA Modernization Blueprint.

Efficient and repeatable usage of these critical methods and standards will help maintain the integrity of the system as components are developed, revised, and integrated. If critical methods and standards are executed poorly, it can result in solutions which:

- do not meet the Modernization Blueprint requirements;
- are not correctly constructed; and/or
- result in cost and schedule overruns.

This model will provide the framework and guidance to successfully address these potential issues.

## **1.2 Scope**

This Critical Methods and Standards approach applies to all information systems and related organizational engineering activities that might affect the achievement of the SFA Modernization Blueprint effort, including hardware, software, organization and process. This approach includes high level views that focus on what is needed to control the development or enhancement of a business capability. In particular, the Capability Analysis stage is defined to meet the specific needs of SFA. The goal is to have the SFA organization focus on defining new or modified business capabilities which deliver significant, measurable value to the organization. Other stages are defined through entry and exit standards/criteria because they are the responsibility of vendors. The stages are:

- Capability Analysis,
- Capability Release Design,
- Capability Build and Test, and
- Capability Deployment.

The testing methodology/approach is not explained in detail in this document but is the subject of the System Integration and Test Approach document. This is illustrated in the following diagram:

## Task Order 2 Prescribed Processes

	Planning & IRB	Capability Analysis	Capability Design	Capability Build and Test	Capability Deployment	Operation
<b>Critical Methods &amp; Standards</b>	<ul style="list-style-type: none"> <li>Gain IRB Approval</li> <li>Develop Business Case</li> </ul>	<ul style="list-style-type: none"> <li>Collect &amp; Manage Req'm'ts</li> <li>Define New or Modified Business Capability</li> <li>Architect &amp; Configure Release</li> <li>Document Release for Vendor Bids</li> </ul>	<ul style="list-style-type: none"> <li>Complete Stage Entry Criteria</li> <li>Complete Stage Exit Criteria</li> <li>Collect &amp; Manage Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Complete Stage Entry Criteria</li> <li>Complete Stage Exit Criteria</li> <li>Collect &amp; Manage Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Complete Stage Entry Criteria</li> <li>Complete Stage Exit Criteria</li> <li>Manage Requirements</li> </ul>	
<b>System Integration &amp; Testing</b>		<ul style="list-style-type: none"> <li>Define Test Requirements</li> <li>Define Metrics</li> <li>Plan Testing</li> </ul>	<ul style="list-style-type: none"> <li>Establish Test Environment</li> <li>Define Test Scenarios</li> <li>Build Scripts</li> <li>Identify Test Resources</li> <li>Define Test Data</li> </ul>	<ul style="list-style-type: none"> <li>Modify/Test Configuration</li> <li>Complete Unit Testing</li> <li>Complete Integration Testing</li> <li>Complete Pilot Testing</li> </ul>	<ul style="list-style-type: none"> <li>Test for performance</li> <li>Test for user acceptance</li> </ul>	
<b>Deployment</b>		<ul style="list-style-type: none"> <li>Plan Deployment</li> <li>Plan Deployment Timeline</li> <li>Obtain SLA/OLAs</li> <li>Build Cost Model</li> <li>Define training requirements</li> </ul>	<ul style="list-style-type: none"> <li>Identify Staff Requirements</li> <li>Plan Ops Readiness Test</li> <li>Design Physical Environment</li> <li>Design Training</li> </ul>	<ul style="list-style-type: none"> <li>Plan Site Layout</li> <li>Upgrade Infrastructure</li> <li>Complete Procurement</li> <li>Conduct Training</li> <li>Recruit &amp; Select Resources</li> </ul>	<ul style="list-style-type: none"> <li>Deploy Infrastructure</li> <li>Deploy Application</li> <li>Deploy Business Processes</li> <li>Transition Workforce</li> </ul>	
<b>Post Deployment</b>				<ul style="list-style-type: none"> <li>Obtain and Manage Vendors</li> <li>Develop Ops Mgmt Process</li> </ul>	<ul style="list-style-type: none"> <li>Verify Operational Readiness</li> <li>Finalize Performance</li> <li>Verify Maintenance Processes</li> </ul>	<ul style="list-style-type: none"> <li>Manage operational system</li> <li>Manage physical equipment</li> </ul>
<b>Configuration Management</b>	<ul style="list-style-type: none"> <li>Review/Approve CR</li> <li>Review/Approve releases</li> <li>Define CM library structure</li> </ul>	<ul style="list-style-type: none"> <li>Initiate &amp; Monitor CR</li> <li>Analyze Change Request</li> <li>Store info in CM repository</li> </ul>	<ul style="list-style-type: none"> <li>Initiate &amp; Monitor CR</li> <li>Analyze Change Request</li> <li>Store info in CM repository</li> </ul>	<ul style="list-style-type: none"> <li>Check Out/Check In CIs</li> <li>Install Configuration Items</li> <li>Configure Application</li> <li>Close CR</li> <li>Store info in CM repository</li> </ul>	<ul style="list-style-type: none"> <li>Establish Release Baseline</li> </ul>	<ul style="list-style-type: none"> <li>Track problem report</li> </ul>
<b>Program Management</b>	<ul style="list-style-type: none"> <li>Define program plan</li> <li>Define resource/ vendor plan</li> <li>Define quality plan</li> <li>Define risk mgmt plan</li> </ul>	<ul style="list-style-type: none"> <li>Report Status</li> <li>Manage Risk</li> <li>Manage Quality</li> </ul>	<ul style="list-style-type: none"> <li>Report Status</li> <li>Manage Risk</li> <li>Manage Quality</li> </ul>	<ul style="list-style-type: none"> <li>Report Status</li> <li>Manage Risk</li> <li>Manage Quality</li> </ul>	<ul style="list-style-type: none"> <li>Report Status</li> <li>Manage Risk</li> <li>Manage Quality</li> </ul>	<ul style="list-style-type: none"> <li>Report Status</li> <li>Manage Risk</li> <li>Manage Quality</li> </ul>

### 1.3 Current State of Methodology

SFA does not follow an enterprise-level methodology. Although Method/1 has been purchased, it has not yet been implemented within SFA. The different “stove-pipe” groups have different methodologies and each practice their previously developed processes. The SFA methodology focus spans the entire life cycle, but up-front activities (i.e., planning, analysis, and design) and back-end activities (i.e., testing) are the main areas of SFA involvement.

There are two different approaches to requirement management and testing that were identified by two representative development teams: EDEExpress/EDEExpress Suite and TIV WAN. Both of them are briefly described here.

#### 1.3.1 EDEExpress/EDEExpress Suite Development Methodology

The EDEExpress/EDEExpress systems development team produced five documents in the support of requirements management and development.

#### Gather Detailed Requirements

The requirements began with the previous release as the baseline due to the annual release schedule. For example, year group software 1999-2000, is used to baseline year group software 2000-2001. Inputs for requirements come from three areas:

- Legislation (i.e. governmental requirements from Capital Hill, budgeting efforts in Congress, etc.),
- Configuration Management (i.e. software bugs), and
- User group/focus group (i.e. state colleges, combination schools, 3<sup>d</sup> party servicers, trade schools, etc.)

From all the inputs, a requirement tracking summary (RTS) report is created using the CaliberRM tool. This report uniquely identifies the requirement, description, date, source, resolution, status, allocation, and functional specification number. An issue can be classified as Global (G), Printing (P), Help text (H), or other. An issue can also be from a previous release (year group software 1998-1999). Everything on this report gets approved/deferred/dropped with justification at workgroup meetings. These meetings contain SFA, development staff, contractor management, contractor development staff, and subcontractors, if applicable. Subject matter experts will be brought in if detailed knowledge is necessary before making a decision.

### **Create Customer Requirements Document**

The requirement tracking summary report is then used in creating a customer requirements document (CRD) using the CaliberRM tool. The customer requirements document contains only the accepted issues. This document details the breakdown of the requirement tracking summary issue and enables the developer to understand the code change necessary. The customer requirements document is created by contractor development staff and the requirements analysis team.

### **Create Customer Allocation Document**

The customer requirements document then is used to support the creation of the customer requirements allocation document (CRAD) using the CaliberRM tool. This document contains a unique identifier, source, description, functional specification number, status, and allocation (Software change group (S), Documentation change group (D), etc.) It specifies who will be required to make changes to support the new requirement.

### **Create Functional Specification Document**

The customer requirements allocation document then is followed by the production of the functional specification document (FS) using the CaliberRM tool. This document provides a description of the requirement and the steps behind it. It explains the functional navigation for the new requirement.

### **Create Technical Specification Document**

Once the functional specification is completed, the technical specification document (TS) is created using the CaliberRM tool. This document details the changes that need to be made to the code. This document will contain changes that apply to functional specifications (FS) and

software change requests (SCR). Addendum are typically produced for the technical specification due to software change requests. Software change requests come from the PVCS Tracker tool and identify software change requests due to application bugs.

### **Build and Test Solution**

Contractors begin development once the technical specification is drafted. At the completion of the technical specification for a module, the contractors will perform unit testing. Once completed, a version is released to SFA and passed onto the customer service representatives (CSR) to test. The customer service representative test provides a user perspective. Macro International performs the systems integration test. Macro International puts together a systems test plan, test procedures, and matrices package. The package is reviewed and inventoried by SFA. All shortfalls are fixed and the plan is started. Macro shares test data sets and files with SFA for their acceptance testing. This test ensures that the customer requirements document requirements are met. SFA's acceptance test coincides with the beta test. The beta test allows 6-9 schools to work with the new software application. If any problems are discovered via the customer service representative test, Macro test, SFA acceptance test, or the beta test, the issues are logged into the PVCS Tracker. A risk analysis is performed on the issue to determine if the issue will be fixed or documented on the requirement tracking summary for the next release. Regression testing is then started to test every fix approved for the current release. After regression testing is passed, integration testing begins. This testing requires detailed coordination with interacting systems.

### **Conduct Production Readiness Review**

The completion of the integration test signals the start for the Production Readiness Review (PDR). This is a statistical review of the entire RM to testing process. Management will review statistical test data and suggest further areas to investigate before releasing the application.

### ***1.3.2 TIV WAN Development Methodology***

The TIV WAN systems development team, unlike the EDEExpress/EDEExpress Suite team, is less paper productive in its requirements management.

### **Gather Detailed Requirements**

Requirements are determined from focus groups of customers (i.e., PC, LAN-based, and Mainframe-based schools). Suggestions are given for enhancements at the informal focus group meetings. Suggestions are also taken from customer service forms. All this information is placed into the PVCS Tracker tool.

### **Review and Approve Requirements**

The requirements are then placed within the tracking summary report and organized by their TIV WAN area. A formal SFA meeting is held to approve the requirements on the report. Budget, security, customer service, and legislation all influence the decision of making a requirement request an approved change.

### **Create Functional Specifications Document**

The approved requirements are then used to build a functional specifications document. This document supports the developers. It identifies what the software needs to do to satisfy the requirements. This document must be signed-off by SFA before NCS will begin any development.

### **Design, Build, and Test Solution**

As development progresses, NCS will provide system flow charts and test cases to SFA. The flow chart shows where reports are generated and files are transmitted. The test cases are reviewed with SFA before testing begins. After approval of test cases and the development of the code, NCS begins testing. NCS will send reports of test matrices to SFA. The reports identify the test owner, project manager, and quality assurance (QA) representative.

SFA pays particular attention to the PC-based code. SFA reviews the code for documentation and performance aspects. Although, NCS provides additional documentation for the mainframe solution, it is reviewed by SFA but not checked as vigorously. The mainframe documentation includes a data dictionary, algorithm check, systems flow chart, JCL catalog, coding standards, etc.

The application is rolled out to the customer service representatives once SFA approves the testing process of NCS. The customer service representatives are used as testers. At the same time, SFA begins the acceptance testing of the PC-based application. The application is also distributed to 5-10 selected schools for beta testing. The tests are focused on the software, the interfaces, the install/uninstall capability, etc. If an issue is found, a document is faxed to the NCS developers. The problems are fixed and if time and availability exist, a new beta version is released to the schools. If the application has already been released, a patch is implemented on the web site for users to download.

## **1.4 Document Organization**

This document includes the primary responsibilities of a critical methods and standards program in the system life cycle process and details the concept of fulfilling these duties.

- **Section 1:** Begins with a description of the context within which Critical Methods & Standards is to be performed on the SFA project. This is followed by an overview of the general approach that will be used.
- **Section 2:** Describes the critical methods. In this document, we focus on the Capability Analysis stage. This is the stage SFA has called Requirements Management.
- **Section 3:** Describes the exit and entry criteria standards.
- **Section 4:** Describes the implementation plan for the critical methods and standards.
- **Section 5:** The critical methods and standards appendix of useful material to support implementation.



## **1.5 Document Development Process Overview**

This section identifies the processes used in developing the SFA Requirements Management Approach.

### **1.5.1 Documents Governing Methodology and Standards**

This approach is consistent with the concepts presented in:

- Andersen Consulting Business Integration Methodology Guide v2.0
- “Business Integration Methodology” white paper on the Business Integration Essentials Knowledgebase

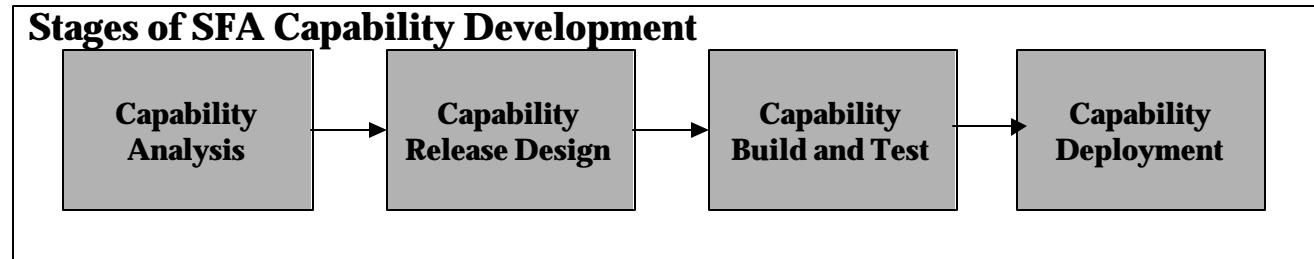
### **1.5.2 Advisors**

The following organizations and individuals were source of information in writing this approach:

- SFA Enterprise IT Management: Denise Hill
- SFA TIV WAN Development: Keith Wilson & Yolanda Brooks
- SFA E-Commerce Application Development: Mike Rockis (EDEXpress/EDEXpress suite)
- SFA Method/1: Bill Bush
- BIM Tier 2 Manager for the Support Center Carolyn Dissoubray
- BIM Help Desk representative: Kimberly Lynn Holston
- Andersen Consulting Methodology Architect: James P. Behling

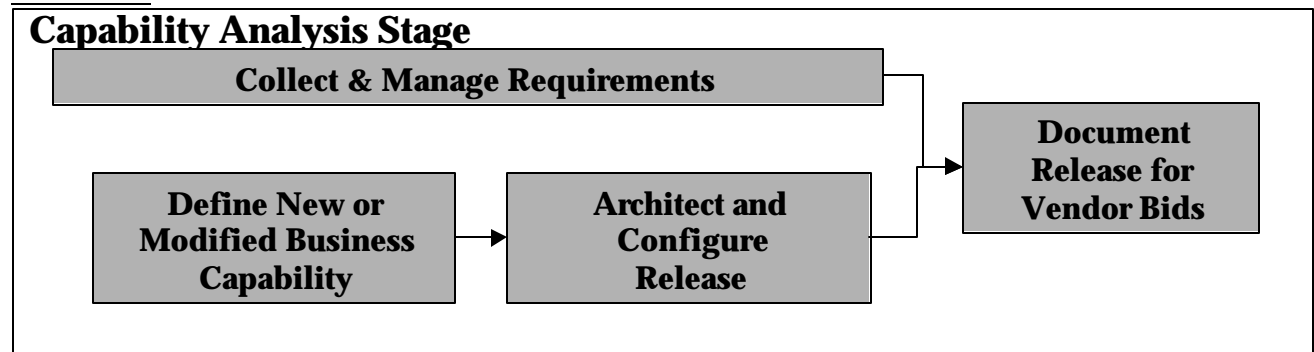
## 2 Critical Methods

This section describes critical methods involved in the Capability Analysis Stage. This stage is the critical method necessary to develop and manage requirements for a business capability prior to contracting out the design, build and test, deployment, and/or operations of it. The relationship of this stage to the previous stages is show in the figure below:



### 2.1 Capability Analysis Stage

#### Schematic



#### Description

In the Capability Analysis Stage the next release of business capability is defined, architected and released to vendors for bids. This purpose of the stage is to describe the outcome, the future business capability, in such a way as to allow the vendor to co-own the outcome, not just the specifications. The stage deals with necessary software for the business capability but also the necessary changes to organization, business process, performance management, and technology.

The key outputs of this stage are:

**Business Capability Requirements** – The requirements developed before, during, and after this stage of the process that form the basis of the release. These will be stored in a requirements management tool.

**Business Performance Model** – The performance specifications of the business capability when delivered. It includes performance measures and targets.

**Business Process Workflow and Dialog Workflow** – The flow of work and the flow of system interaction are described in order to fully describe the business practices of the new business capability.

**Variability Matrix** – The variability matrix describes the transactions and conditions that are handled in the business capability. It is a detailed scope definition for the capability. For instance, a given business process may or may not include Pell Grants. The Variability matrix would illustrate this coverage. Also, if there are certain exceptions conditions for loans or grants that exclude them from this business capability they are also illustrated. Any new or especially difficult conditions can be highlighted in the matrix.

**New Component Configuration Items (CI's) and Modified CI's** – The identification, specification, and requirements tracing to new and modified configuration items is a key control point for SFA. SFA owns the architecture, that is, the structure of the system and SFA should specify new configuration items such as business components or data entities. No longer should contractors deliver systems with arbitrary internal structures. The structures should match the master plan of SFA.

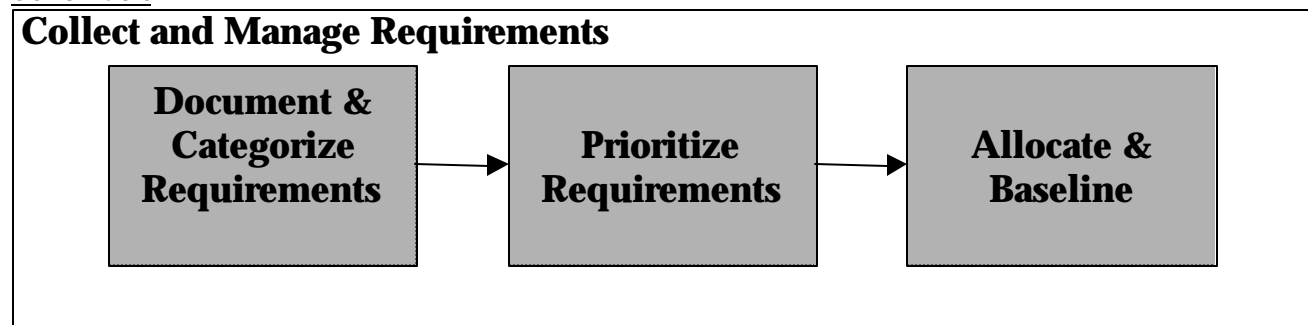
**Request for Proposal (RFP) and Request for Information (RFI)** – The Capability Analysis stage ends with the choice of a contractor to deliver the business capability. The RFP or RFI are key elements in expressing the requirements of the business capability so that a suitable contractor can be chosen.

The level of detail within this stage varies. Significant variables include degree of change to the capability, commitment of sponsor and stakeholders, existing knowledge capital in functional and technical domains, and urgency. Where less is understood, more detail is required. A maintenance release which is primarily software focused and to be awarded to the organization that wrote and operates the software may require less detail. When there will be significant changes in the business processes, organizational structure and responsibilities, and/or infrastructure technologies, it is better to develop the requirements to a lower level of detail to more fully assess impact and increase the accountability of all parties on the desired outcome.

A deliverable matrix listing all deliverables by Activity and Task appears in appendix 5.9 Deliverable Matrix.

### ***2.1.1 Collect and Manage Requirements***

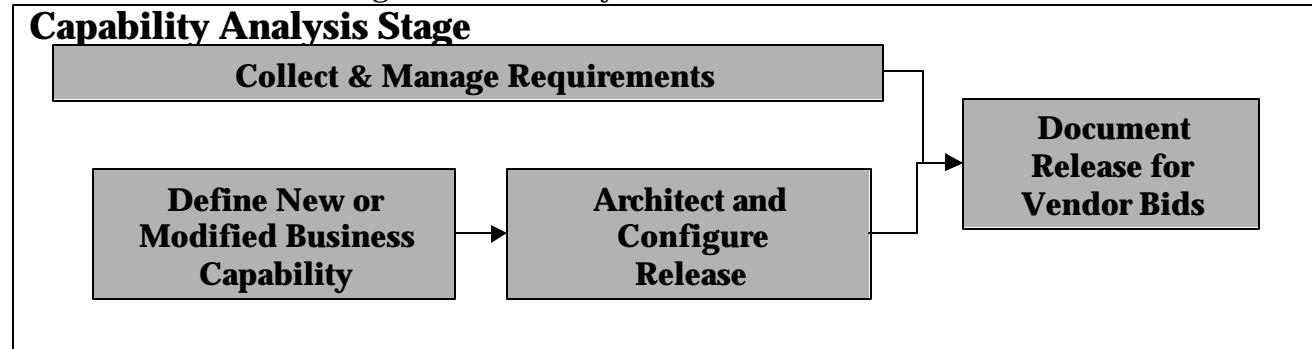
#### **Schematic**



#### **Description**

The purpose of the *Collect and Manage Requirements* activity is to gather all requirements into the repository and prioritize them for the current release. The activity is executed concurrently with

the *Define New or Modified Business Capability* and *Architect and Configure Release* activities (see next diagram). As these activities are being conducted, new requirements will be uncovered that should be collect and managed in this activity.



During this activity, the team will collect requirements generated during the analysis of the business capability. All the requirements will need to be documented, prioritized and allocated to either the new business capability release, or a future release. One of the objectives of this activity is to capture all of the requirements, even those that might not be included in the release. The exclusion of requirements should reflect conscious decisions when configuring the release.

#### **2.1.1.1 Tasks**

##### **Document and Categorize Requirements**

The first step is to gather the Business Capability Requirements for the new or modified business capability. The requirements are often a result of the definition of the features and functions necessary to successfully build, test, deploy, and operate the business capability. The requirements may also have been accumulated during the development, deployment or operation of the previous release of the capability.

Business Capability Requirements should be assigned to the following categories:

- Business Process,
- Human Performance,
- Application,
- Physical Environment, and
- Technology Infrastructure.

The Business Capability Requirements will provide detailed requirements to the contractor about **how** the capability should operate. These requirements also will drive an accompanying deliverable which is the Capability Performance Model (i.e., the definition of **how well** the business capability should operate).

##### **Task Deliverable:**

- Business Capability Requirements

### **Prioritize Requirements**

As the requirements are identified, categorized, and decomposed, they need to be prioritized. Specifically, they need to be prioritized based on the contribution to the performance of this release of the business capability. The performance targets are defined in the *Define Business Capability Performance* task of the *Define New or Modified Business Capability* activity. During this task, the teams will determine which requirements are essential to the release, borderline for the release and which are excluded.

One technique that can be used in this task is Software Quality Function Deployment (QFD). Software QFD is a quality management technique that links requirements to business needs. These techniques provide a robust approach to defining the business requirements, the benefits they expect from the application, and the priority of the requirements to meet these expectations. The project team uses these priorities to focus their best efforts and deliver value to the organization. See Appendix Section 5.1, Quality Function Deployment for more details.

#### **Task Deliverable:**

- Business Capability Requirements, updated to reflect priority

### **Allocate and Baseline Requirements**

As the release configuration is finalized, it is necessary to finalize those requirements which will be included in the release and those which will be excluded. Those requirements which have been deferred can be considered for another release. Those which have been included need to be identified as belonging to the baseline set of requirements which the contractor must meet in order to deliver the business capability.

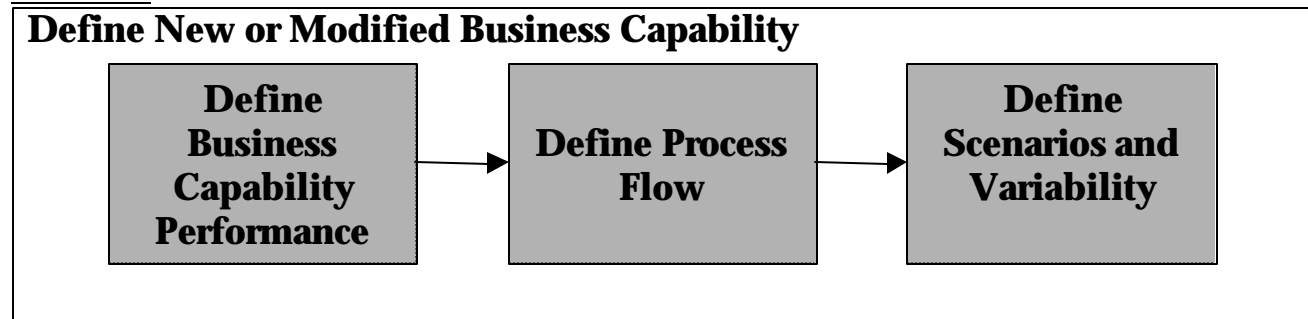
In addition, this is a good point to verify that the Business Capability Validation Conditions created in the *Define Scenarios and Variability* task of the *Define New or Modified Business Capability* activity are still valid and complete before going through the contracting process.

#### **Task Deliverables:**

- Business Capability Requirements, updated to reflect release scope
- Business Capability Validation Conditions, updated to reflect release scope

## ***2.1.2 Define New or Modified Business Capability***

### **Schematic**



## **Description**

Use this activity to define the performance specifications for the business capability and the business practices that the business capability will follow. The scope of the business capability is also described in this activity.

### **2.1.2.1 Tasks**

#### **Define Business Capability Performance**

The Business Performance Model is a central deliverable for the measurement of business processes and capabilities. This model is the instrument for ensuring the delivery of the benefits planned for in the Business Case resulting from the IRB approval process.

The capability analysis team needs to understand the operating vision for the business capability and high-level performance targets. These are specified by the SFA channel(s) that owns or shares the business capability. This will enable the team to design metrics which directly contribute to achieving the high-level performance objectives while maintaining operability of the business capability.

The business performance is defined for each release of the business capability. This consideration allows SFA to immediately measure the capability upon its first release rather than waiting for the completion of the program. Using the customer service example: the first release of the business capability may be able to reduce average wait time from 45 seconds to 30 seconds. A second release may include additional capability that reduces the wait to 20 seconds. In this case, the 30 second metric would be what the business tries to manage to until the second release, where that measure would fall to 20 seconds.

There are two parts to the Capability Performance Model, the Business Performance Model Metrics and Definitions, and the Business Capability Measurement and Communication Procedures. The Metrics and Definitions can be structured by including:

- **Introduction** -- Describes the current performance of the business capability. Identifies the basis of current performance and areas to improve. Identifies the overall performance objectives and describes the targets and measures in the following.
- **Performance Target** -- Documents the performance targets to be achieved. Includes tolerances for the target where possible.
- **Performance Indicator** -- Documents the performance indicators that support the performance target. The structure below should repeat for each metric defined.
- **Performance Metric** -- Details the indicator into one to many metrics that give detailed formulations on how to measure the performance target. These metrics often take the form of a formula, but that form is not required.
- **Metric Definition** -- Defines the metric and its contribution to performance.

See the Appendix Section 5.2, Business Performance Model Metrics and Definitions for guidelines and a sample.

The Business Performance Model should focus both on results and actions. When defining metrics for the Business Performance Model, assume an outcome-focused approach. Consider:

- What is the outcome of the business capability?
- What are the necessary performance levels to ensure that this outcome is consistent?
- How can we ensure that the performance levels are consistent?

A useful way of developing and measuring outcome-focused capabilities is to define metrics from a customer and supplier perspective. The customer and suppliers can be both internal and external to the company. The outcome of a business capability has a customer. Consider:

- What does the customer expect?
- What performance levels are necessary to ensure that the customer's expectations are met?
- How can we ensure that the performance is at the necessary levels to meet customer expectations?

A business capability also has a supplier. Consider:

- What is expected from the supplier?
- How can we ensure that the supplier input is consistent?

The purpose of a Business Performance Model is to follow the progress of implementation and to have an impact on people's *behavior*. If the business metrics are on the business manager's agenda or five year performance plan, they will have a strong impact on people's behavior and priorities.

The second part of the Business Capability Performance Model is the Business Capability Measurement and Communication Procedures. This section contains the performance indicators for the business capability as well as the description of how to collect and process the measurement data required by the Business Performance Model. The section also includes information about who should receive information, in what format, how frequently, and so on. See Appendix Section 5.3, Business Capability Measurement and Communication Procedures for guidelines and a sample.

While developing the Capability Performance Model, now is the time to make the first draft of the Business Capability Validation Conditions. This deliverable will document the conditions and their measurement criteria by which the success of the business capability will be measured.

**Task Deliverables:**

- Capability Performance Model
- Business Capability Validation Conditions

### **Define Process Flow**

After the Capability Performance Model has been defined, the next task is to define the process flow for both core and supporting processes and activities for a business capability release. It is at this point that any reengineering of business processes will take place. It includes the definitions of relationships between core and supporting processes, activities, and tasks, and the metrics associated with them. Any current supporting processes or activities that do not contribute to meeting the Capability Performance Model are possible candidates for elimination. The level of effort for this task depends on the amount of new process design. A key success factor will be the use of a combination of key individuals from the sponsoring organization and subject matter experts to help design teams define the detailed tasks and workflows.

One of the major products of this task are the Business Process Workflows. A workflow diagram is a graphical representation of a process or activity showing the sequence of lower level activities and tasks, with associated decision points. This enables the contractor to understand the information and work flow of the capability. To design workflows, identify:

- Triggers and outcomes
- The movement of inputs/outputs
- The sequencing of tasks
- Hand-offs/ handover points for tasks
- Where and how all resource types are deployed.

For processes that are really performed by the computer systems, the functional flow of the process should be diagrammed as a workflow. The team should make sure not to spend time designing the technology but should specify the functionality.

Developing Business Process Workflows can become a very involved task. For more detailed information on Process Modeling in general, see Appendix Section 5.4, Process Modeling.

Use the Capability Performance Model as the driver for defining metrics for processes, activities, and tasks. Decompose performance targets into operational performance requirements which can be applied at task and workflow levels. Use the high level performance targets and measures identified in the Capability Performance Model to design detailed workflow level metrics. The metrics should be defined in such a way so that they drill down the hierarchy of metrics. The correct measurements should:

- Encourage the right behavior in an organization;
- Provide the best technique to produce accurate results.; and
- Eliminate poor performance.

Concurrent with the definition of the Business Process Workflows, the team should be developing the Dialog Flow Diagram. This deliverable illustrates the user's interaction with the application via one or more windows.



These diagrams can be used at different levels of details, varying from simple names of windows with arrows, to definitions of functionality attached to both the windows as well as the arrows. In this stage of the system development life cycle the dialog flows should be at a functional level. They should not specify specific data or user interface look and feel. See Appendix Section 5.5, Sample Dialog Flow for more details on this deliverable.

**Task Deliverables:**

- Business Process Workflow
- Dialog Flow Diagram

**Define Scenarios and Variability**

The next task begins the effort to clearly define the scope of the business capability. The objective of creating a Workflow Variability Matrix is to specify the scope of the release by brainstorming and summarizing the factors that cause the business processes to vary. It is important to note that the Workflow Variability Matrix is an intermediate deliverable. The ultimate objective is to ensure the scenarios and workflow diagrams are complete and account for the full breadth and depth of business process variability. The technique used to accomplish this objective is not as important.

Identifying the scenarios help fully describe the scope of the solution. Scenarios are specific instances of data passing through a process with descriptions of how the data is handled at each step of the way. They are real and concrete. They talk about a specific school and a specific student in a specific program. Each scenario should document one path through a business process. Together, all of the scenarios for a business process should span the variability that is in scope. Scenarios may apply at any level of the business process model. For example, information at the process, activity, or task level could be validated. It is important to perform this session with business experts in the area discussed. However, it is also extremely important that the scope is solid. If there are any questions about scope, return to the business process design and resolve any open issues.

At the conclusion of this task the scope of the capability should be clear and illustrated by examples of how all types of transactions will flow through it. By doing this, test conditions have been created for the business capability because the scenarios will be concrete and “testable.” Also, the contractor will have clear examples to understand the requirements and scope of the work.

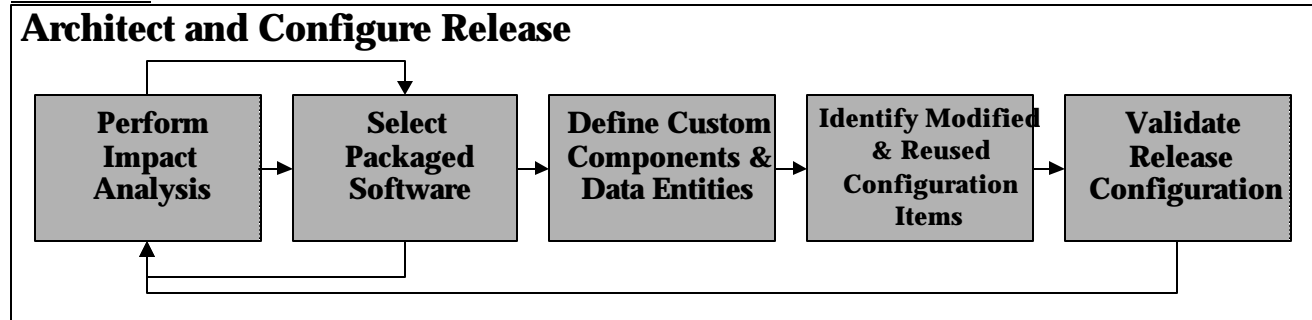
To see a more detailed discussion regarding the creation of scenarios see appendix 5.6 Scenario Development Technique and appendix 5.7 Using a Variability Matrix.

**Task Deliverables:**

- Workflow Variability Matrix and scenarios

### **2.1.3 Architect and Configure Release**

#### **Schematic**



#### **Description**

This activity defines the structure and the contents of the release. Up to this point the process has focused on requirements. Here they are allocated to specific configurable items and any architectural decisions that SFA wants to dictate in the release are made.

##### **2.1.3.1 Tasks**

#### **Perform Impact Analysis**

In this task, the team analyzes the impact the planned changes have on the configuration items that make up the current capability. The goal is to identify all elements that will change in order to place them under configuration control for the release. A side benefit is that many requirements will be uncovered through this analysis. Also, this effort will give insight into the cost of the new capability based on what is known at this point in the process. The team should assess the impact on human performance, business processes, technology and infrastructure, and physical environment.

When analyzing the human performance, the team should cover skills needed and training requirements, necessary recruitment and succession planning, changes to organization structure, changes to reward system due to new skills, location or working hour scheduling, redundancy and transfer of employees

Additional skills and training requirements will be dependent on the level of current skills and the skills required. The impact should consider all types of changes in technology and in the organization, due to new business capability requirements (e.g., multi-skilled due to integration of tasks, new software etc.).

It is possible internal training will not be sufficient and a complementary addition in recruitment and succession planning will be necessary. The reason for this can be significant changes in the jobs, expansion of the business (or part of the business) or locations.

Changes in the organization can be required due to the impact of the business capability requirements. Areas that can be impacted are new jobs, new reporting structure, new

responsibilities and accountabilities, a new organization structure as well as a new organization culture. The cost and time associated with the above will be an important input when creating the Release Impact Analysis.

The team should evaluate the impact on information technology and infrastructure, having the prime focus within capability boundaries but considering the impact and/or constraints outside the boundaries of the capability. Consider the impact of technology infrastructure changes. Infrastructure and information technology (IT) are usually the backbone of a number of different processes making it very clear that implementing changes needs to adopt a wider view.

The introduction of a “new” IT Application can have effects within business processes and on its interfacing processes. For example, the option of a new payroll application can have an effect on both administration/personnel and IT maintenance processes. Moreover, the introduction of an enterprise wide solution can have a dramatic effect on every single business process in the sponsoring organization.

Equally the introduction of a new network topology, or a change to a Internet architecture, are good examples for illustrating that infrastructure changes can have impact on business processes.

The changes in technology and infrastructure are very important since they are an integral part of the operations of the sponsoring organization, requiring specific skills, and absorbing large amounts of capital. Therefore, the effects need to be assessed and consolidated accurately by considering all related business processes.

Each impacted entity should be loaded into the configuration management tool and denoted as part of the release.

**Task Deliverable:**

- Release Impact Analysis

**Select Packaged Software**

The goal of this task is to identify the potential vendors that have the ability to satisfy a subset of the Business Capability Requirements by using their packaged software products. The team should first select two or three packaged software vendors as finalists for detailed evaluation. After evaluating the vendors, the team will have a cursory view of the market and confirm whether or not a packaged software approach is a better solution than the custom application approach. If there is no appropriate software package, the team should move on to the next task, *Define Custom Components & Data Entities*.

If there are appropriate software packages, the team could develop a request for proposal (RFP) in order to bring in and evaluate the software packages. It is common today, however, to forgo the

RFP altogether. In this instance, the project team may submit an RFI (Request for Information) to the vendor which includes the Business Scenarios or simply attach a cover letter to the Business Scenarios. RFPs are becoming less common for a number of reasons including the following:

- The marketplace is more mature.
- Business Scenarios are becoming more accepted.
- Vendors tend to stretch the truth regarding their answers.
- Vendors do not have their best people complete the criteria matrix, or they assign several people to the matrix.
- Too much time is spent scoring the criteria instead of analyzing the answers.
- Focus on the business issue is lost.

The team should next evaluate and select a software finalist. To accomplish this, they should conduct vendor demonstrations, contact vendor references, and even conduct site visits. As the team compares the cost and benefits of the vendors, they should begin to negotiate terms with the finalist.

During the evaluation, the team should develop a Requirements Traceability Matrix. Use of a Traceability Matrix is a technique where a matrix is developed of all requirements cross referenced to the configuration items, the designs, code, tests, and deployment deliverables which implement the requirement. What is accomplished is a cross check that all Business Performance criteria are directly linked to one or more requirements, and that for each requirement there is a direct link to a configuration item. This facilitates the identification of missing requirements, additional requirements (scope creep), and the identification of specifications not contributing to the business capability performance goals. At this point in time, the goal is to identify those requirements which are met by the software package, without customization.

The last step is to finalize the Vendor and document the justification. This justification will be reviewed with the sponsors and stakeholders at the end of the activity.

**Task Deliverables:**

- Software Selection Justification
- Requirements Traceability Matrix

**Define Custom Components and Data Entities**

Once the packaged software has been selected and its functionality documented against the Business Capability Requirements, it is time to define the custom components. These custom components will complement the packaged software and are necessary to implement the business capability. Certain classes of components are important to SFA. In the world of component based development, the business components are key application elements that SFA should define for contractors. For instance, SFA would not want three contractors to define different Loan components. This would make reuse very difficult and eliminate one of the central benefits of component based development. These custom components can be:

- Application programs,
- Business components,
- Technical architecture components

As these components are identified they should be added to the Requirements Traceability Matrix and placed under configuration control.

Data entities should also be managed by SFA. The vendor responsible for implementation should not be able to create and define data entities. SFA owns the corporate data model and should define any significant data entities. This way the introduction of a new package or system does not compromise the integrity of the corporate data.

**Task Deliverables:**

- New, Modified, and Reused Component Configuration Items
- Updated Entity Relationship Diagram
- Requirements Traceability Matrix updated with new custom components

**Identify Modified & Reused Configuration Items**

This next task is a continuation of the previous task, except it is expected that there will be some existing components which can and should be reused from other releases. As existing components are identified, the Requirements Traceability Matrix should be updated. It must be made clear to the Vendor that they should reuse the existing components. This reuse should save the SFA both time and money. The Vendor should reuse application software and application architecture components, as well as infrastructure components. For example, they should be required to use system software that is already licensed (e.g., database software). Also, they should use common date and print routines that have already been developed. Architecture components are pre-built and funded on the basis of their reuse in applications. If contractors build their own components instead of reusing the SFA components then maintenance will be harder and the payback will be lost.

**Task Deliverables:**

- New, Modified and Reused Component Configuration Items
- Requirements Traceability Matrix updated to reflect reused of existing Configuration Items

**Validate Release Configuration**

The final task is to validate that the release configuration is complete. The Requirements Traceability Matrix must be complete, identifying how the requirements are to be met, i.e., package, custom, or reused components. An estimate of the resources required to deliver the business capability and its related cost should be refined. Note that it may also be necessary to renegotiate the scope, schedule, or resource commitments of the release if revised effort levels and target dates are significantly different from previous plans. If this is the case, be sure to identify any resulting effects on the Business Case. Go back to the *Perform Impact Analysis* tasks if there is a significant change.

This is also when the review and, if necessary, revision of the Business Capability Validation Conditions occurs. These conditions will prove extremely import in measuring the effectiveness of the Vendor in delivering the business capability.

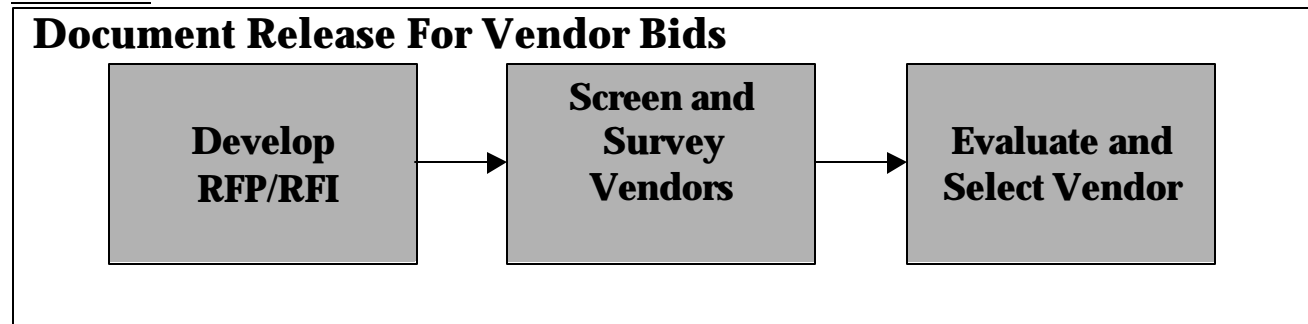
Finally, prior to preparing for vendor bids, the team should solicit sponsor and stakeholder approval to proceed with the Capability Release Design Stage.

**Task Deliverables:**

- Updated Business Capability Validation Conditions
- Updated Business Case

**2.1.4 Document Release for Vendor Bids**

**Schematic**



**Description**

The final activity is to contract out the design, and possibly the build, test, and deployment stages. These steps should stay similar to what SFA has done in the past. Just how many stages of the development cycle are contracted out will depend, among other things, on the scope of the change, specificity of the analysis, and the familiarity of the Vendor with the business process, human performance, and technical infrastructure of SFA.

The important change suggested here focuses in on what the Vendor is actually contracted to do. SFA should contract out for the delivery of a *business capability*. The performance of the Vendor should be measured primarily on the success of the change, i.e., the performance of the new or modified business capability. This performance is documented in the Business Capability Validation Conditions.

**2.1.4.1 Tasks**

**Develop RFP/RFI**

The team will first need to decide if they will go through the formal RFP or RFI route. Criteria to use to help determine the route should include:

- Degree of competition within specific market,
- Size of budget, and

- Familiarity with vendors.

The RFI contains general questions (based on screening criteria) to send to vendors so that background information may be obtained about their company and its services. The general outline of the RFI should include:

- Project Approach and Status
- Vendor Selection Schedule
- Project Overview
- Evaluation Questions

The RFP should be more detailed than the RFI. Also, if going the formal RFP route, the team should schedule vendor meetings which are based primarily on business capability requirements.

**Task Deliverables:**

- Request for Proposal or Request for Information

**Screen and Survey Vendors**

This step is the normal process SFA goes through to evaluate and survey vendors.

**Task Deliverables:**

- Short list of vendors

**Evaluate and Select Vendor**

The last step is to evaluate and select the vendor. Again, this step should not change much except for the importance of the vendor's ability to deliver the business capability.

**Task Deliverables:**

- Task Order

## **2.2 Testing Process**

The objective of the testing process is to ensure that the delivered SFA system and related processes satisfy the defined functional, technical and quality requirements. The testing process includes all activities required to conduct thorough and accurate tests of system parameters, customizations, interface modules, conversion programs, reports, and business processes. The permeating philosophy is to build testing into the development process, rather than making testing the last step before production. This philosophy extends beyond traditional testing approaches by ensuring that all major deliverables completed during a SFA system implementation project will be verified and validated throughout each stage of the system development life cycle.

The details of the testing methodology will be detailed in the System Integration & Testing deliverable.



## 3 Standards

### 3.1 Purpose

The objective of stage containment is to identify and correct defects at their source before they are passed on to a subsequent stage of development or testing. Problems become exponentially more expensive and difficult to fix the later in the development life cycle they are detected. By concentrating on containment, the cost of fixing defects can be decreased and a quality product delivered. Stage containment is a project management style driven by the need to minimize the number of problems from development to implementation. From a business perspective, stage containment principles can be used to minimize the risk of having a project progress too far that does not have the potential of delivering measurable and meaningful business benefits to the SFA organization.

Entry and exit criteria are used to implement stage containment. These criteria are predefined standards that deliverables must meet before exiting from one stage and entering another. This includes criteria covering quality, function, technology, and procedure.

Entry and exit criteria improve quality by defining an agreement between the supplier (in the most general terms) and the recipient of a deliverable. A team handing work off to another team must fully complete their exit criteria, while the receiving team verifies that the work meets their standard entry criteria.

*Entry criteria* describe the preconditions that exist before a particular phase or stage of work should begin. These preconditions include: prerequisite deliverables, appropriate levels of executive commitment and ownership, and levels of knowledge. Entry criteria provide a basis for assessing the potential risks of starting work without these prerequisites in place. This helps inform the program manager and team leaders of the gaps that exist between their situation and the baseline assumed by the methodology.

Note that typically the supplier's *exit criteria* for a stage can be considered as part of the entry criteria for the next stage. In most cases, the added significance of the entry criteria is the commitment from the sponsoring organization to begin the next stage. Clearly that commitment is not an exit criteria from the previous stage.

Stage containment is achieved when a deliverable is accepted from a stage only after it meets that stage's exit criteria. Any deliverable failing to meet its criteria is returned to the stage that developed it.

Ideally, all criteria must be met before a deliverable is passed from one stage to another. However, in practice, a project should establish a set of minimally acceptable criteria. Should a team decide to accept a deliverable that does not fully meet its entry criteria, this discrepancy should be recorded. For example, if a component is ready for product test but has one known bug, it can be

promoted in order to keep testing on schedule; this decision is knowingly made by both the customer and the supplier. It is up to the two teams involved to reach an agreement as to when a new version of the deliverable is to be delivered. This situation should be the exception, not the rule.

The major stages described here in this Critical Methods and Standards document are:

- Capability Analysis,
- Capability Release Design,
- Capability Release Build and Test, and
- Capability Deployment

For each one of these stages, the Stage Purpose and Stage Overview are described. In addition, the Entry and Exit Criteria are **summarized**. **Detailed** Entry and Exit Criteria are provided in the Appendix. The detailed Entry and Exit Criteria have not been customized to any specific project type but have been customized for SFA. They should be reviewed for applicability by the project manager when beginning a project. For instance, if a project will not result in changes to the organization structure, then exit criteria related to organization design would not be part of the project's exit criteria.

## **3.2 Stage Entry & Exit Criteria**

### **3.2.1 Capability Analysis Stage**

#### **3.2.1.1 Stage Purpose**

The work in the Capability Analysis stage concentrates on defining the business requirements, performance targets, and configuration necessary to define and commit to a specific implementation and release plan. The information gathered during this stage focuses on a single business capability, describing it to the level of detail needed to finalize the delivery releases, define the specific requirements, and resolve implementation issues.

#### **3.2.1.2 Stage Overview**

This stage begins with an organization's commitment to a delivering program to meet specific performance and operational objectives. These objectives should be bound by specific expectations regarding cost and schedule. The stage ends with a defined release plan and a detailed scope definition for a business capability; if the solution is contingent on the use of packaged software, then preferred software should be identified and included in the costs. The organization commits to that plan based on the business requirements, the Business Performance Model, and the approaches used to develop the capability releases. Finally, the stage ends with the selection of a vendor (or vendors) to continue with the next stage, if not all of the stages.

#### **3.2.1.3 Summary of Entry Criteria**

The Capability Analysis Stage must begin with the key stakeholders having approved the Business Case that supports the benefits stream, in either in quantitative or qualitative terms, the development of a new or enhancement of an existing business capability. They must formally express their acceptance and support for the plans to proceed with the more detailed analysis the capability. Also, they should have set expectations regarding the expected cost and schedule of the entire effort to implement the business capability. The IRB process calls for these conditions to be met.

Finally, the quick wins related to the business capability should be estimated and incorporated into the program plan, with a high priority to staff and execute. These quick wins are characterized as changes (generally small) to the business process(es) or standard policies which when implemented have a significant, immediate impact to the organization.

#### **3.2.1.4 Summary of Exit Criteria**

At the end of the Capability Analysis stage the team should have finalized the Business Capability Requirements, developed the Capability Performance Model, updated the Business Case, and completed the Business Capability Validation Conditions. The Business Process Workflows and Dialog flows should be developed and account for the full breadth and depth of business process variability. Then the next stage, Capability Release Design, plans and estimates should be finalized; in addition, the estimates for work in the later stages of the Capability Development process, as well as for subsequent releases of the business capability should be refined. Any vendor software to be used as part of the solution should be completely evaluated and contract

terms finalized. Any work to be contracted out should be finalized through task orders. Finally, the team then needs to obtain commitment from the sponsoring organization, based on the revised plans and estimates, to proceed with the Capability Release Design stage.

### **3.2.2 Capability Release Design Stage**

#### **3.2.2.1 Stage Purpose**

The work performed in the Capability Release Design stage concentrates on defining the implementation details of the changes to human performance, business process, and technology that are needed to improve business performance. This stage is conducted once for each release of the business capability. The work in this stage results in the design of new organizations, competencies, learning materials, business processes, applications, and technology infrastructure necessary to achieve the Business Performance Model. These designs form the basis for the detailed estimating and scheduling of the work required in the Capability Release Build and Test stage.

#### **3.2.2.2 Stage Overview**

The activities within this stage address the design and integration of the elements with the business capability release. The overall design process is often iterative, as teams will continuously revisit their designs in order to resolve integration issues and tune the overall capability to meet the Business Performance Model. Teams should start off by designing the new business processes, a step begun in the Capability Analysis stage, and define how they will interact with the workforce (skills), applications (application interaction), and physical environment, based on the Business Capability Requirements and the Business Performance Model.

The team focusing on human performance should then define structures for managing human performance. These structures define what is expected of people who participate in the business capability operation, the required competencies for people to perform the business capability, and how the performance is managed and sustained. The team can then determine the training that helps people make the transition to their new roles, as well as to define online help text, procedures, job aids, and other information that they can use during the actual work.

The team focusing on business processes should identify the Business Capability Requirements that will be delivered through an application. This entails analyzing elements of the current Business Process Design, and understanding the business processes, the requirements from the stakeholders (e.g., employees, application users, customers, suppliers, sponsors), the usability requirements, the required performance of the product in the organization, and the required performance support for the stakeholders.

The team focusing on the physical and technical environment should select and design the:

- Execution Architecture,
- Development Architecture,
- Operations Architecture, and

- Physical network and computing platform.

Once these tasks are complete, the team should then ensure that the development, execution, operations, physical network, and computing designs are integrated, compatible, and consistent.

The last major activity of the stage is to ensure that the capability release design supports the realization of the Business Case and Business Performance Model. This final step in the design process represents a rigorous mechanism to investigate and simulate the design before sign-off of the capability release.

### **3.2.2.3 Summary of Entry Criteria**

Before the Capability Release Design Stage should begin, the required sponsors, executives, and other stakeholders should have authorized the stage effort, including changes in scope to the business capability, the stage budget, and the stage schedule. It is critical that the capability release scope, as documented in the previous stage, is defined in sufficient detail to proceed with design. Also, these requirements should not conflict with the strategic direction of the organization or the business area being enhanced by the business capability. Finally, the cross-release dependencies and implications of the current release are identified and deemed acceptable.

### **3.2.2.4 Summary of Exit Criteria**

The team needs to confirm that the new business process designs and the definition of their interactions with the workforce (Skills), applications (Application Interaction), and Physical Environment have been verified and validated that they meet the Business Capability Requirements and are internally consistent. The team should ensure the designs are verified against the requirements by facilitating stakeholder sign-off of the application, user interface, database, and data conversion design deliverables. Also, the team needs to ensure that all the essential action and resource plans for acquisition and deployment of the physical environment transformation are complete. Finally, the team should review all of the designs as a whole to ensure that the capability release design supports the realization of the Business Case and Business Performance Model.

The Capability Release Build and Test stage plans and estimates have been finalized and committed to by the project team; these will become the baselines against which the stage will be managed. Any package software should be installed. The technical architecture designs should all be complete, with all the third party development and execution architecture products installed and integrated. Ideally, the custom architecture components would not only be designed but would be well into the build activities. The team should identify any gaps in its integration with the existing architecture components and address these gaps quickly. Finally, the development environments and associated tools (configuration management, testing, compilers, etc.) and resources (PCs, servers, storage) should be in place and operational.

The estimates for work in the Capability Deployment stage, as well as for subsequent releases of the business capability have been refined. Finally, the commitment from the sponsoring

organization to proceed with the Capability Release Build and Test stage of work should be obtained.

### **3.2.3 *Capability Release Build and Test Stage***

#### **3.2.3.1 Stage Purpose**

The Capability Release Build and Test stage concentrates on building the business capability elements required for a single capability release.

#### **3.2.3.2 Stage Overview**

The delivery teams are responsible for the detailed design and creation of new processes, facilities, learning systems, performance support, application systems, and technology infrastructure components necessary to implement the new capability. These elements are then tested and implemented within a pilot environment.

#### **3.2.3.3 Summary of Entry Criteria**

First, the Capability Release Build and Test work effort, completion schedules, and estimates should have been approved by the sponsoring organization. Though the Capability Release Build and Test stage requires additional individuals with different skills (e.g., programmers and training developers), ideally, individuals who have worked on the Capability Analysis and Capability Release Design stages should work on Capability Release Build and Test teams. This ensures continuity throughout the development of the capability.

In addition, the Capability Release Build and Test work environment should have been established. This environment primarily includes the technology infrastructure, but could also include a work environment or office setup which will support the capability.

Also, necessary test plans, test conditions, and expected results should be complete.

#### **3.2.3.4 Summary of Exit Criteria**

The capability has been built and tested. Test results have been reviewed and approved by SFA. The technology has been piloted. The deployment approach is approved by SFA. Business continuity plans are in place. Training material and data conversion are ready for deployment.

This is the last time to before deployment to check that the business case for the release is still valid and will achieve acceptable results. Key performance indicators to measure the capability should also be aligned with the capability. The stakeholders should have sufficient confidence in the capability release to decide that the next step is to deploy it.

### **3.2.4 *Capability Deployment Stage***

#### **3.2.4.1 Stage Purpose**

The Capability Deployment Stage is conducted to transition the organization to the new business capability.

#### **3.2.4.2 Stage Overview**

This stage includes the activities used to transform the human performance, business process, and technology elements that are required to establish/deploy the capability in accordance with the Business Performance Model. The deployment stage is repeated for each deployment unit, which is the organizational or geographic unit (for example, location, role) that will receive the business capability.

#### **3.2.4.3 Summary of Entry Criteria**

The sponsoring organization should be committed to the deployment of the business capability. This commitment should involve senior management, line management, and staff.

Confidence to deploy when there is a deployment plan with a trained deployment team to execute it. The foundation of this plan is a deployment package that contains:

- All elements of the application;
- All elements of the business policies and procedures;
- All elements of the physical environment;
- All elements of the technology infrastructure; and
- All elements to transition the workforce.

In addition, the risks associated with deployment of the business capability have been addressed through a mitigation plan.

#### **3.2.4.4 Summary of Exit Criteria**

The key elements of the deployment package have been properly installed and are ready to be activated. The monitoring of the planned progress has verified the stability of the deployment unit in using the deployed business capability release, and is ensured that a “ready to release” state has been reached. If there are any legacy systems, they have been removed from the operations and management processes.

## **4 Implementation Plan & Compliance Monitoring**

### **4.1 Implementation Plan**

To bring the SFA organization to higher, more mature level of performance in the area of Requirements Management, the implementation plan addresses the following challenges:

- The need to quickly deliver results while balancing the need to grow a requirements management capability,
- The need to define requirements which deliver measurable value in terms of new or revised business capabilities,
- The unfamiliarity of people to working in Integrated Product Teams,
- The variability in the way different groups prepare their requirements,
- The ability of the different groups to assimilate change in process while still delivering, and
- The potential use of a new tool(s).

The requirements management process that has been outlined in this approach should be the first priority for SFA. Implementation of such significant change, however, is not a trivial task. Before beginning a change program, a detailed, executable plan which has executive support should be created.

#### **1. Set Performance Goals for IPTs**

Assess the current state of the IPTs and their ability to follow this requirements management process. Describe the enabling characteristics and capabilities of a high performing, successful IPT. Set performance objectives for the IPTs over the short term (1 month), medium term (3-4 months) and long-term (6-9 months).

#### **2. Develop High-Level Overview**

The purpose of this task is to develop a high-level overview of the critical methods and include all of the steps of the IPT development methodology. The first step is to develop a high level overview of the processes. This overview should be used to communicate the scope and responsibilities of the processes to the CIO community and also to the channels and CFO. The second step is to integrate the Investment Review Board and Independent Verification and Validation processes. Finally, the various reviews associated with the processes need to be identified and agreed to.

#### **3. Identify Process Tasks**

The purpose of this task is to identify the detailed tasks for each of the process steps. These tasks need to identify who has primary leadership for performing the task and who, if anyone, will also participate in the task.

#### **4. Identify Process Reviews**

The purpose of this task is to identify what reviews of the IPT process need to take place. These reviews will be at critical points in the project where specific deliverables should be reviewed. The main points of the review and the review criteria should be identified.

#### **5. Develop key deliverable templates**

The purpose of this task is to develop initial (i.e., MS Word or MS Excel-based) deliverable



templates for potential use by the IPT's. The templates should also include descriptions of the fields to be completed.

6. Conduct Training and Communication

The purpose of this task is to communicate to the IPTs the process, along with the associated tasks and templates. They will be trained in each one of the steps of the process, focusing on the deliverables produced and the reviews conducted.

7. Develop plan to integrate development tools into IPT process

The purpose of this task is to develop a plan to integrate the use of tools in the IPT process.

The use of tools generally has a significant impact on process tasks and deliverable format. In addition, deployment of new tools require implementation and training to ensure effective use.

## **4.2 Compliance Monitoring Approach**

To ensure that the Critical Methods and Standards are followed, SFA should implement compliance monitoring. To monitor compliance, SFA should create a position to perform the following activities:

- Review IPT deliverables for adherence to standards. Any deviations should be documented as either detracting from the quality of the deliverable or improving the quality of the deliverable.
- Review the IPT workplans and status reports for adherence to standards. The workplans should indicate what the IPTs have done and will do. The status reports should also cover what has been done and the outcomes of the tasks.
- Report to the responsible CIO executive the compliance status of each of the IPT's. It would then be the responsibility of this executive to take the appropriate action.

## 5 APPENDIX

### 5.1 Quality Function Deployment

QFD provides the project and business representatives with a clear definition of value based on measuring the relationship between what the business representatives want (their needs) and the capabilities of the software (its functional requirements). Value is defined in terms of the software's capabilities and its support of the business needs. QFD is a well known and established technique for requirements management. It is also complex. The approach is recommended and introduced here but a full explanation is beyond the scope of this document.

QFD provides a systematic approach to identifying business value and associates functional requirements to achieve that value. The project team and business representatives can actively manage the application's development process to maximize the business value of the software.

Identifying and managing value throughout the software development process involves focusing the project team's efforts on the aspects of the software that matter most to the customer. QFD enables this approach through a process of defining and illustrating value to the project team. The high value areas of the application are communicated to the project team through a prioritized set of functional requirements.

Software QFD consists of a number of techniques and work objects that discover and communicate the relationships between business needs and application requirements by addressing the following issues:

- Who are the business representatives using the application? Who must the project satisfy in order to be successful?
- What do the business representatives say regarding the application? What are the problems and opportunities the project team can address to be successful?
- What application capabilities address the business representatives' problems and opportunities?

QFD uses many of the seven management process quality tools including matrices, tables, affinity analysis, and hierarchies to derive the answers to these questions. QFD applies the Analytic Hierarchy Process (AHP) to produce priorities. This process determines priorities through straightforward math and the business representatives' relative judgments.

#### 5.1.1 The QFD Process

QFD augments the project's other approaches and techniques for applications development. It supports the discovery, classification, and prioritization of business needs and requirements. It is most effective when used on the subset of requirements that are associated with the greatest value. In this regard, QFD does not remove the need for full exploration and delineation of the application requirements.

The primary techniques associated with this process include affinity analysis (or the KJ Method™) and the Analytic Hierarchy Process™ (AHP). Affinity analysis produces the users' view of the structure of the business needs before the project structures the business needs hierarchy. The AHP provides a robust approach for prioritization and is used throughout QFD.



The majority of the activities associated with software QFD occur in the Capability Analysis stage where the project's scope and value are determined. QFD is organized around six deliverables that are displayed in Figure 2.0. These deliverables structure the information gathered through QFD.

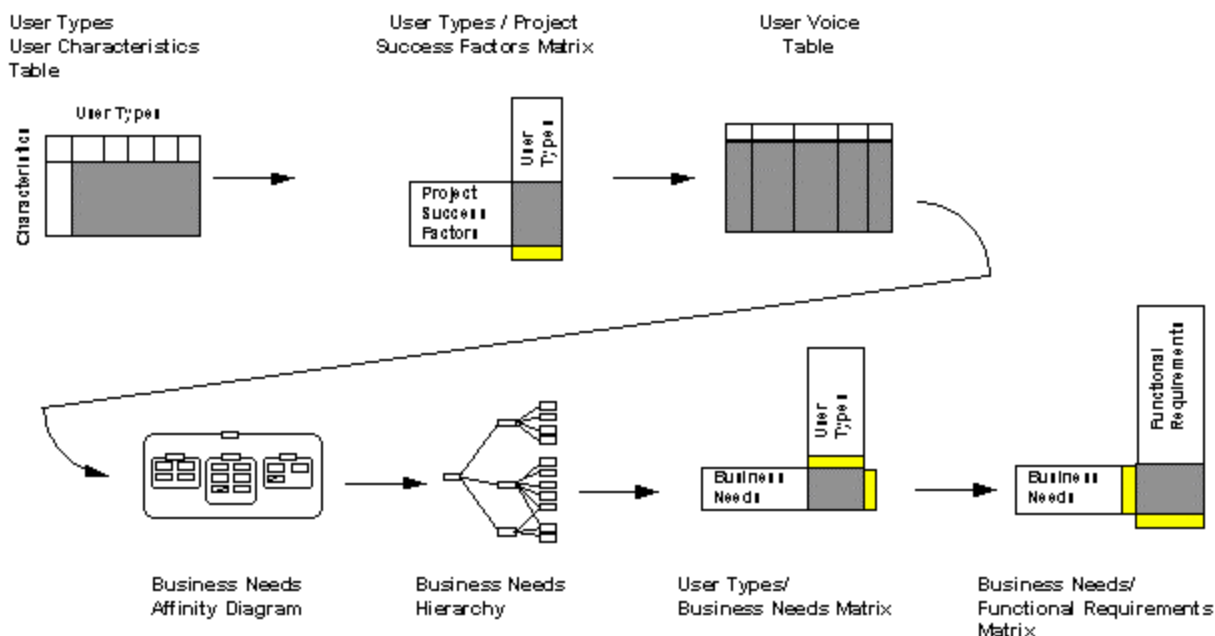


Figure 2.0 Roadmap

#### 5.1.1.1 Determining User Types and Characteristics

Quality Function Deployment's focus on business satisfaction is reflected in the first activity of determining the user types. This activity produces the User Types/User Characteristics Table and the User Types/Project Success Factors Matrix. These work objects identify whom the project must satisfy in order to be successful. The users are described and their importance is weighted by the project success factors. This information provides a guide regarding each user type's involvement in the development process.

#### 5.1.1.2 Evaluating Requirements

The User Voice Table offers a tool for organizing the business representative input collected during requirements gathering. The User Voice Table evaluates business representative comments and requirement statements to sift out business needs, functional requirements, quality attributes, project requirements, and other statements. The analyst uses the voice table in analyzing the users' comments and points. The business needs identified on the voice table are carried to the next step of the process.

#### 5.1.1.3 Defining Business Needs

The users' business needs represent problems and opportunities the application could address. These needs require thorough examination in order to determine the structure and level of business benefits the project could deliver.

The first step in this process is to understand how the users view the business needs. The Business Needs Affinity Diagram provides an initial view of this structure that is further refined and

finalized in the Business Needs Hierarchy. The hierarchy supports the users and project team members in assessing and valuing the business needs and their benefits to the organization.

#### **5.1.1.4 Assigning Business Needs to User Types**

Once the business benefits have been structured and evaluated the project team needs to assess the users' satisfaction with the business needs. The User Types/Business Needs Matrix provides a tool to predict the level of business representative satisfaction with the application based on the business needs it will focus upon. The level of business representative satisfaction provides a means of prioritizing the business needs based on their ability to fulfill the user types.

#### **5.1.1.5 Aligning Requirements to Needs**

The organization receives value from the new application to the extent the application addresses the business needs. The Business Needs/Functional Requirements Matrix prioritizes the application's functional requirements based on their contribution to the business need.

Prioritizing requirements according to business value rather than to each other enhances the users' and project's ability to establish and manage project scope. This, coupled with the extensive business representative involvement required by QFD, creates an empirical and open process of determining and managing scope.

#### **5.1.1.6 Managing Value**

Prioritizing requirements alone does not guarantee the application's value. A focused development process uses this definition of value in managing their development process. Project management communicates this information so the team members know what matters most to the users and where their best efforts can maximize the application's value.

Defining where the value is in the application also serves as a guide to the downstream activities in the application's development process. The priorities and value define where the project team should devote their scarce resources to doing their best work. This is particularly helpful when the application is being tested or prototyped.

### **5.1.2 Performing QFD**

Quality Function Deployment is generally conducted by a small team (one to six people) during the planning and initial analysis activities. The exact size and duration of the QFD activities are determined by the initial scope and requirements of the project.

One goal of QFD is to understand the users in order to focus the project on satisfying them. This requires extensive business representative involvement and input throughout the process. The team and process often facilitates the users in defining and measuring the project's scope and benefits.

Users react well to software QFD for several reasons:

- The process is open and defined to allow the users to see where and what type of input is needed.
- The process produces numerical priorities and data that allow the users to see the results of their input and comments.
- The business representative input directly feeds into and guides the application's development process. Users can see that their time and effort make an impact on the project.

These factors contribute to the users' involvement in the QFD process. However, software QFD can create a contract between the project and the business community. This contract arises from the definition and prioritization of the requirements. This raises an expectation that high priority requirements will reflect the project team's best efforts.

### ***5.1.3 QFD and Joint Application Development***

The structured nature of QFD along with its requirement for business representative involvement fosters its use in Join Application Development (JAD) sessions, an efficient and interactive approach to obtain the required business representative input. QFD and JAD work best with a small group of users who understand not only the business and project charter but also the QFD process. They should be capable of making judgments that bind their groups.

Individual steps within the QFD process can be performed in a half or one day JAD session. This is particularly the case in gathering and deciphering business representative requirements as well as the affinity analysis of the business needs. See the Joint Application Development Guidelines job aid for more details.

### ***Summary***

Software Quality Function Deployment provides a powerful set of techniques for the application development team. It clarifies and records information concerning who the application's users are, what they want, and why they want it. This information constitutes the dynamics of both functional scope and the Business Case. The project needs to understand these dynamics in order to build an application that meets the customers' needs and therefore delivers business value.

## 5.2 Business Performance Model Metrics and Definitions

### Guidelines

It is very important to understand the organization that will be measured both broadly and in detail. The risk of defining incorrect performance indicators cannot be underestimated. Defining metrics is a difficult but very important task. Keep in mind that “What’s measured gets done” and “People will consistently react on how they are measured.”

When defining measures, make sure they are properly reviewed before communicating them. Consider the following:

- Can you gather the data that you want without a major cost? High costs could be due to a very time-consuming exercise, or to a major program change required to gather the right data.
- Are all the performance indicators in line, not conflicting?
- Do you follow and meet all requirements defined in the business capability requirements?
- Will the receiver of the follow-up understand the performance indicators?
- Do you want a fast-reacting or slow-reacting performance indicator, or both? A longer measurement period will normally give a smoother development of the performance indicator.
- How many performance indicators should be defined? If you send too many performance indicators to the same receiver, there is a risk that they will lose the focus.
- Will the performance indicator measure the right thing every time? (For example, a change in the product or the service mix can have a major impact on the performance indicator.)

The Business Performance Model also should define those actions that are vital for the success of the program or project. An important action in the application could be the updating of all the lead times from suppliers.

The total number of performance indicators and important actions can be significantly high, and the number will probably increase as the program and the projects develop. The performance indicators will not be equally important, therefore define priorities. A balanced scorecard approach can be used to prioritize and present the key performance indicators.

### Volume

One per Business Capability

#### 5.2.1 *Sample Business Performance Model: Order and Delivery Management*

##### Introduction

XYZ’s Operating Vision calls for:

Increased response time to customer orders and inquiries to increase customer satisfaction, retention, and growth in orders

Improved Sales and Profits, particularly by increasing the European business etc.

### **Performance Targets and Indicators**

<b>Objective</b>	<b>Target</b>	<b>Baseline</b>	<b>Indicator</b>	<b>Metric</b>	<b>Definition</b>	<b>Stakeholder</b>	<b>Business Value Impact (from Business Case)</b>
Improve delivery time of customer orders from 10 days to 5 days from receipt of order	Deliver 99% of orders within 5 days of order receipt. Cannot deliver beyond 7 days.	XYZ currently delivers 10 days from receipt of customer order	Percentage of orders that meet the 5 day delivery objective	(Number Orders where days < 5 / Total Orders) * 100  (Number Orders where days > 5 and supplier problem / Total Orders) * 100	Defines the percentage of orders where the order was delivered in 5 days.  Defines the percentage of orders delivered over 5 days where there were supplier problems.	CEO and staff, customers	Grow revenues by \$1 million/ year  Lower inventory costs by \$400,000 / year
Increase European Sales and Margins by 8-10% in Europe per annum	Increase European Sales and Margins by 8-10% in Europe per annum	XYZ's European sales growth is currently at 1% per year	Sales volume  Margin	((Sales this year - Sales last year) / Sales last year) * 100  Gross Profit / Sales	Defines the percentage increase in sales volume  Defines gross margin received	CEO and staff	Increase European profits by 12%  Lower operating costs by 5%

### **Sample Description**

The sample shows one of many ways in which this deliverable may be presented. The sections below describe the basic structure that can be used for capturing the metrics at business capability level. Adjust the formatting for the actual deliverables as necessary, through the use of tables, graphs, charts, or other formatting options.

### **Introduction**



Describes the current performance of the business architecture or business capability. Identifies the basis of current performance and areas to improve. Identifies the overall performance objectives and describes the targets and measures in the following.

**Performance Target**

Documents the performance targets to be achieved. Includes tolerances for the target where possible.

**Performance Indicator**

Documents the performance indicators that support the performance target. The structure below should repeat for each metric defined.

**Performance Metric**

Details the indicator into one to many metrics that give detailed formulations on how to measure the performance target. These metrics often take the form of a formula, but that form is not required.

**Metric Definition**

Defines the metric and its contribution to performance.

## 5.3 Business Capability Measurement and Communication Procedures Guidelines

The capability measurement and communication procedures should consider questions such as:

- What are the key performance indicators?
- Who should do the actual measuring (information gathering)?
- When should the measurement take place (frequency of measurement/variability to the measurement during the program or project)?
- When should it be sent out (frequency)?
- Where should the measurement take place (all over or sample)?
- In what form should it be presented (table, graph)?
- Where should it be presented (project meetings, company meetings, bulletin boards etc.)?
- Who should receive each type of performance indicators and in what format?
- Must confidentiality be considered when it is presented (depends on where it is presented)?
- Can any existing information channels be used (company paper, road shows, groupware software etc.)

When developing this deliverable, make sure that it is reviewed by program or project management or, even better, that they take an active role in developing the capability measurement and communication procedures.

### Volume

One per Capability

#### 5.3.1 Sample Business Capability: Measurement Procedures

KPI/Metric	Responsible Party	Frequency of Updates	Presentation Format	Confidential Performance Indicator	Audience	Distribution Date	Distribution Channels
KPI 1	Purchasing Dept	Quarterly	Graph and Table	Yes	Program mgt. Purch. Dept.	5 working days after quarterly report	Bulletin board, Company paper
KPI 2	Logistic dept. Consulting org.	Monthly	Graph (table)	No	Project mgr. Logistic and purchasing	5 working days after monthly report	Bulletin board, Company paper

<b>KPI/Metric</b>	<b>Responsible Party</b>	<b>Frequency of Updates</b>	<b>Presentation Format</b>	<b>Confidential Performance Indicator</b>	<b>Audience</b>	<b>Distribution Date</b>	<b>Distribution Channels</b>
KPI 3	Purchasing director: Back-up:	Monthly first 6 months and after 6 months quarterly	Table	Yes, the names of the supplier but not the number	dept. Program mtg. and Top mgt.	To be presented at top mgt. meetings	

### 5.3.2 Sample Business Capability: Performance Tracking

<b>KPI/Metric</b>	<b>Baseline Performance &lt;Period&gt;</b>	<b>Current Performance &lt;Month/Year&gt;</b>	<b>Target Performance 1 &lt;Month/Year&gt;</b>	<b>Target Performance... &lt;Month/Year&gt;</b>	<b>Target Performance n &lt;Month/Year&gt;</b>	<b>Comments</b>
KPI 1	22.4 cents/liter	xx.x cents/liter	21.6 cents/liter	21.0 cents/liter	20.5 cents/liter	
KPI 2	32 d. (\$43 m.)		25 d. (\$33 m.)	19 d (\$26 m.)	15 d. (\$20 m.)	
KPI 3	4 very large 25 large		3 very large 20 large	2 very large 15 large	0 very large 5 large	

## **5.4 Process Modeling**

This section provides guidelines, principles, and standards for all aspects of process modeling to document and design processes, activities, and workflows. The job aid covers process modeling, decomposition, documentation and diagramming.

### **5.4.1 Overview**

Business process rests at the center of all organizations. The process defines the activities that must occur in producing results. A comprehensive process model contains a process map (workflow) with the activities that transform inputs to produce results, along with documentation of the business rules guiding those activities and the activities' performance measures.

#### **5.4.1.1 Process Modeling Levels**

Process modeling occurs at many levels within an organization. High-level process models distill the organization's operations into no more than a dozen processes. The International Benchmark Clearing house categorized common business processes for all organizations including: Generate Demand, Fulfill Demand, Plan and Manage the Enterprise, and Develop Products and Services. At this high level, process models define the major concepts an organization works with to create value for itself and its customers.

High-level process modeling however, remains too abstract unless it is supported by successive layers of detail. This detail breaks high-level process models out into finer degrees of specification, ultimately ending with detailed workflow that documents the individual actions of the workforce and information technology.

#### **5.4.1.2 A General Approach for Process Modeling**

Process modeling at any level follows the same approach documented graphically in Figure 1.

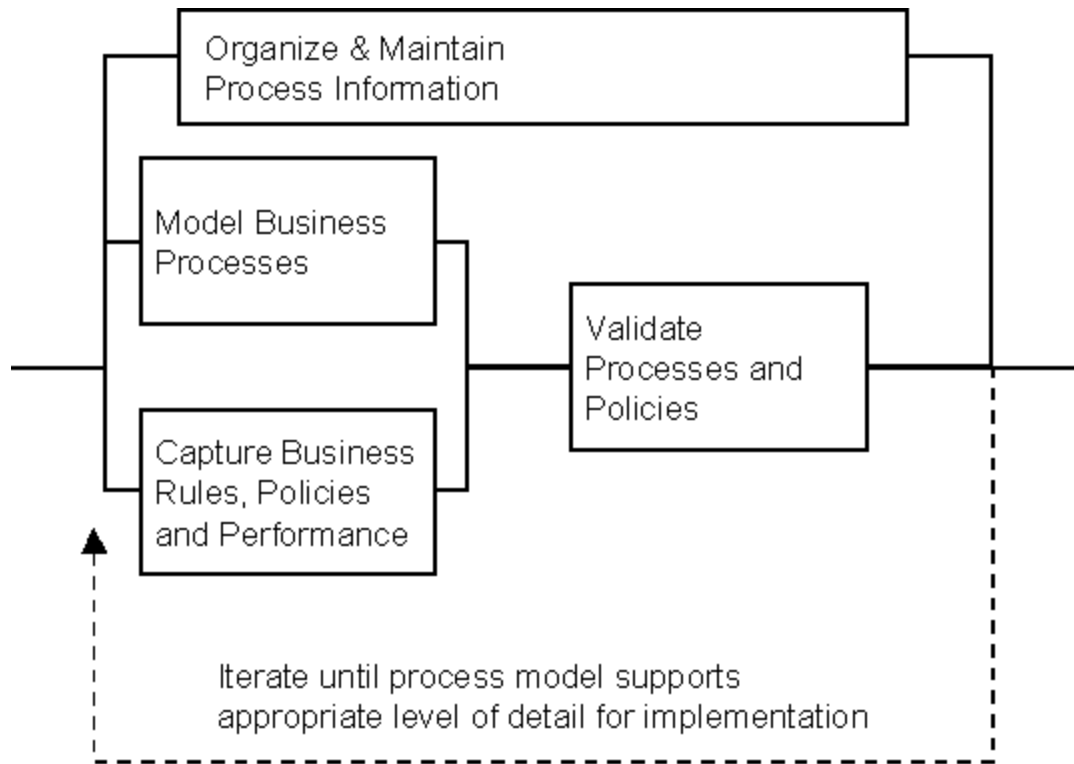


Figure 1. Generic approach for process modeling

- Model Business Processes--Model business processes by creating an end-to-end process model, diagrammed to the appropriate level of detail, including all possible entities, inputs, outputs, workflow routes, rework loops. This results in Process Definition and Process Diagram deliverables that describe the activities contained within the business process. When predictive and statistical validity regarding performance metrics are needed, process simulation models should also be created as part of the Model Business Processes step.
- Capture Business Rules, Procedures, and Performance--This occurs concurrent with process modeling in order to collect the relevant business rules, procedures and performance within the process. Particular attention should be paid to resource and role requirements related to the process as these will be key drivers of a processes operational costs and performance characteristics.
- Validate Processes and Policies--This provides early validation of the process model, policies, and rules to ensure they are complete and contain sufficient detail for their later implementation. The process model should be validated against several criteria including its completeness, accuracy, and ability to communicate the activities required.
- Organize and Maintain Process Information--Process information is dynamic and changes as development teams explore multiple options to meet the required level of business performance. For this reason, most process modeling occurs within modeling tools. Ensure

that any process modeling team understands and makes proper use of available modeling tools.

#### **5.4.1.3 Process Modeling Teams**

The process modeling team is important as process modeling is normally done as a set of facilitated sessions organized around the generic approach illustrated above. The process model has a comprehensive scope covering the organization's activities, resources, business rules, and other aspects. The breadth of the process model requires a wide range of skills and participation involving the development team, business expertise, and process subject matter expertise. Personnel contributing to process modeling teams should have a good understanding of the subject area, process modeling concepts, and the performance objectives of the future business process.

Process modeling involves capturing the complexity of an organization with a consistent set of definitions, symbols, and diagrams. The leader of the process modeling effort, as well as the people using modeling tools should be skilled in the business area and toolset. This is particularly the case when the modeling effort will involve process simulation. The skills and experience of these people enable the team to concentrate on business issues rather than being bogged down in modeling semantics and tools functionality.

#### **5.4.1.4 Static and Dynamic Process Modeling**

Process modeling captures the flow of activities and entities through an organization. Those activities are frequently captured in static diagrams such as workflows and swimlanes. These models are said to be "static" because they do not include real-world uncertainty (i.e., probability distributions) of the elements of the process nor take the cause-effect behavior of the process over time into account. Static process models work best to capture the activities, their dependencies, and business rules of enterprise business processes.

Complex business processes, characterized by numerous activities, business rules, resources, and entity flow, may benefit from the "dynamic" process analysis provided by process simulation. Process simulation adds the elements of real-world uncertainty/variability of process elements and the dimension of time (e.g., to accomplish activities, queuing, etc.) to analyze and predict the behavior of the process. Not all processes need to be simulated or can benefit from process simulation and it is useful to know when this additional technique can and should be utilized.

### **5.4.2 Process Modeling Definitions**

The terminology and components of a process model can be confusing as the team needs to distinguish between process, sub-processes, activities, and workflow. The terminology the methodology uses for process modeling uses the following definitions:

<b>Process</b>	A set of logically related and continuously evolving business activities, or lower level processes, that combine to produce a defined output(s). Processes are triggered by business events (e.g., a customer order).
----------------	---

**Activity**

An Activity is a lower level business process. There can be one or multiple levels of activities (Level 1, Level 2, Level 3, etc.) An activity is a logical collection of lower level activities or tasks which are grouped to execute a specific business function (e.g., Perform Customer Credit Check) before transferring an entity(ies) to the next activity. Activities are designed to produce a particular intermediate output of the process (e.g., Approved or Disapproved Customer Credit). Every Process is made up of activities (i.e., the next level of business process). Activities are performed by individuals or small teams.

**Task**

The Task is the lowest logical or desirable level of process decomposition, i.e., the micro view of a Process. To decompose further would add no value or understanding of the Process in question. For example, Call External Credit Bureau could be a task within the Perform Customer Credit Check Activity. Tasks are performed by individuals or small teams. Tasks (and groups of tasks) are used to understand/develop job requirements and business rules.

**Entity(ies)**

The items or things that flow through, and are worked on by resources in, a business process; for example, customer orders. The process concept of an entity is similar, but not identical to, the data modeling concept of an entity.

**Business Event**

Captured within the entity, a business event is the transaction or condition that initiates a business process. Receipt of a customer order or an account aging over its collection date are examples of business events.

**Business Rule/Policy**

Provide the guidelines and instructions that resource roles are to follow when executing a specific activity/task. The combination of all relevant business rules for a process determine all of the possible routes (business procedures) that entities can take through that process.

**Business Procedures**

Indicate the different “routes” through a process or workflow. The routes are defined by key decision points in a workflow that dictate the tasks executed in response to a decision or condition. For example, there is a different procedure to follow for a new customer and for an existing customer. A decision point will determine whether the customer is new or existing and as a result dictate which business procedure to follow based on this classification.

Each Business Procedure has a unique set of business rules, tasks, metrics, resources and controls. Business Procedures provide input into developing business procedures as a path through the workflow. They

also represent typical execution scenarios for the workflow.

**Process Model**

Is the highest level document containing workflows that provide a graphic description of an end-to-end business process and detailed documentation of the business rules, activity processing times and other descriptive process information. The process model is the core concept within the process documentation. A process model typically involves multiple levels of process decomposition displaying the sequencing and dependencies of activities and tasks, rework and feedback loops, and all inputs and outputs of the process.

Process models are not complete until they have been validated with subject matter experts. Process models are static descriptions of a process and are not analytical by themselves.

**Process Diagram** See Process Model.

**Workflow Diagram**

The key graphical deliverable within the process model. The workflow is a graphic description of a process or activity showing the sequencing and dependencies of the “next-lower-level” processes, activities, and tasks. Most typically, workflow diagrams are produced to illustrate the tasks to execute within a lower-level process or activity. Workflow diagrams can be used to illustrate any level of business process.

### **5.4.3 Process Modeling Standards and Guidelines**

Process modeling needs to be performed with rigor and technique in order to be useful in process analysis and design. A strong process model is one that exhibits the following characteristics:

- Encompasses the end-to-end process flow, covering the complete processing of business events and all business procedures.
- Includes entities that trigger the beginning of the process and that are completed at the end of the process.
- Defines the business rules that determine the various routes entities take through the process.
- Captures the time to work on entities at each process step (i.e., activity or task).
- Accounts for the resources (e.g., HR roles, applications, equipment, etc.) needed to process entities at each step in the process flow.
- Allows for rework loops to occur.
- Allows both sequential and concurrent processing of entities.
- Models the variability that occurs within the model (e.g., in entities, business rules, processing times, rework, etc.).

The approach to building the process models (and collecting other associated process documentation) is critical their accuracy. The following approach should be used:



- Derive it during group facilitation sessions with subject matter experts (SMEs) from the sponsoring organization who work in the process, supplemented with members of the engagement team with experience in working in, or designing, such processes.
- Keep the groups small (three or four SMEs per session).
- Use a facilitator from the engagement team who is experienced in process modeling.
- Focus on building the graphical models, and documenting the business rules that determine model routings, during the initial group sessions.
- Collect the other documentary data and information during a second round of group meetings with the same SMEs.
- Do a final validation of the process models, and obtain sponsoring organization concurrence, during a third and final round of group meetings with the same SMEs.

A process model by itself does not provide sufficient information to document the process and must be accompanied by other text descriptions of the process as described in the Process Documentation Guidelines sub-section. The following Standard Diagramming Shapes should be used when developing either “As-Is” or “To-Be” process models, and whether using process modeling software and drawing process models by hand. The associate quantitative and qualitative information necessary to document the shapes used in a particular instance of a process model are described in a following section.

#### **5.4.3.1 Standard Diagramming Shapes**

The process model captures an organization’s activities and dynamics in the form of pictures and text. The process model represents the business in the form of one or more diagrams that will form the basis for application and human performance designs. Clarity and consistency in the definition and use of diagramming standards is therefore critical to successful process modeling. This next section defines the process diagramming standards for the methodology.

##### **Process Trigger:**



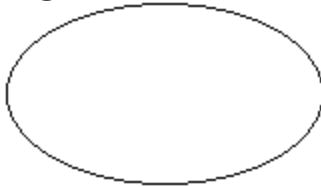
The process trigger (or business event) is the input that initiates the process. It is also the primary Entity (i.e., item or thing) that enters the process to be worked on. For example, for a process “Resolve Customer Query,” the trigger (and entity) would be “customer call/letter.” This would then result in a series of steps to be performed to deal with this call/letter. A trigger differs from an input as it causes a series of process steps to be executed. Triggers can occur at the beginning of a process (e.g., customer calls to place an order), and they can also occur at other points in a process (e.g., a customer calls to dispute an invoice toward the end of an overall order-to-cash process).

##### **End of Process/Final Outcome:**



An output is the “final” result of performing a business process. It signifies that processing of the entity that entered/triggered the process is complete and has achieved the objective of the process (e.g., “goods received by customer to satisfaction”). There can be multiple final outputs for a process (e.g., Loan Application Approved, Loan Application Disapproved and Loan Application Withdrawn are all final outputs for a loan application process). The final output differs from an exit process symbol, as the final output signifies that the process objectives have been met. By contrast, the exit process occurs when the entity being worked on has to leave the process (e.g., for lack of information) before reaching the final output/end of process.

### **High-Level Process:**



A high-level process represents a process at the highest level of detail: for example, “Generate Demand.” The oval process step symbolizes a high-level process (level zero). It should not be used at a low level (i.e., at the level of detailed activity); the process step (see below) can be used for more detailed activities. Examples of appropriate high-level processes include:

- Develop Products and Services
- Generate Demand
- Fulfill Demand
- Plan and Manage Enterprise

### **Process Step:**



The process step represents an action taken by a resource or team of resources during a process. It represents a series of individual steps that are performed in sequence and/or concurrently in order to achieve a business objective. Each step has a unique numeric identifier (see Numbering Standards) which corresponds to the most common order in which the processing is undertaken. Each step can be decomposed to a lower level of detail until decomposition is no longer required. The step will then automatically indicate that a further layer of processing exists. The unique identifier of a lower level step is related to the higher level step.

## Numbering Standards:

The standard for numbering process steps is “deep numbering” (also known as “Work Breakdown Structure”) that identifies the position of the process both in terms of its depth and its sequence. For example:

### 1. Generate Demand

1.1 ---

1.1.1 ---

1.1.2 ---

1.1.3 ---

1.1.4 ---

1.2 ---

1.3 ---

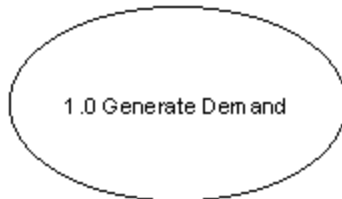
1.3.1 ---

1.3.2 ---

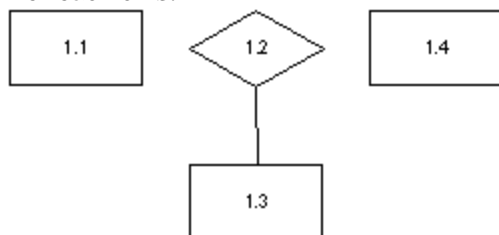
1.3.3 ---

1.4 ---

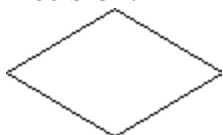
1.5 ---



Beneath this:



**Decision:**



A decision step is used when the entity can take more than one possible route to the next process step/activity. For example: suppose a Customer Order entity can go either to the Check Credit activity or to the Send to Warehouse activity for further processing. The business rule governing

this decision dictates the next step within the process. Suppose the business rule is that---All Customer Orders exceeding X price must receive a credit check, otherwise they are to be sent directly to the warehouse for fulfillment (i.e., skip the credit check step). Business policies and business rules provide the guidelines used by resources (e.g., HR roles, applications, etc.) to make these decisions along the process route taken by each entity. Decision steps typically used only at the lowest (i.e., task) level of process decomposition.

**Common Process Step:**



This represents a series of steps that are performed in more than one process (for example, Update Personal Details). Common Processes are represented on the Process Flow by a common Name and Identifier.

**Exit Process:**



A Process may have a number of “exits.” The symbol for Exit Process should only be applied when the entity being worked on cannot reach the intended final output from this point in the process. Example: in an insurance application process, the business representative wishes to register a new customer for insurance; this process requires personal details from the customer. The HR role resource collects the customer’s name, address, etc., but fails to enter a National Identity Number. Since registration of the entity cannot be completed without a National Identity Number, the entity will have to exit the process at this point, and will not be able to end the process because its final output (a new registered customer) cannot be achieved.

**Input:**

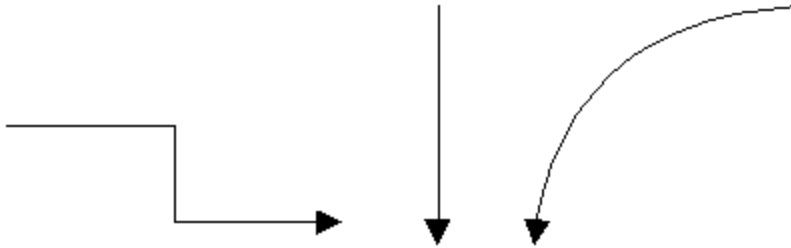
An input is a tangible representation of a product or piece of information that is required to perform a task. Only additional input to the task is captured (i.e., if the input has been recorded in the previous task, it should not be graphically represented in the current task). Refer to the section “Customizable Palette” below to see how inputs can be tailored and graphically represented.

**Output:**

An output is a tangible representation of a product or piece of information that results from carrying out a step: for example, a report that is produced upon completion of a step or a telephone answer to a customer inquiry. Outputs are tangible, while outcomes are intangible (e.g. the customer responded that he/she was satisfied). In addition, process outputs can be either

physical (e.g., a printed invoice) or non-physical (e.g., a telephone response). Refer to the section “Customizable Palette” to see how the inputs can be tailored and graphically represented.

**Connectors:**



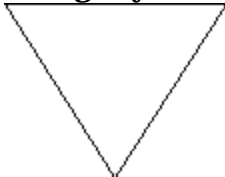
Connectors are the “routes/paths” that link process or decision steps, and can be straight or arced. In some processes, concurrent/parallel processing may occur. This would require a step to have multiple connectors (with each connector going to a step). Also, at levels of process decomposition higher than the task level, where the decision shapes are not used, multiple connectors should link successive process steps when multiple paths are possible.

**Customizable Palette/Symbols:**

The interim inputs or outputs of the process can be represented on the workflow. The customizable palette provides a way to represent these items graphically and to tailor the symbols used: for example, it is possible to import bitmaps to represent common interim inputs and outputs such as a telephone call, a letter, a computer printout, etc.

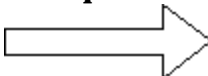
Other Possible Symbols to Use:

**Storage Symbol:**



The storage symbol can be used to illustrate when entities/items are being temporarily stored as opposed to being in a queue.

**Transportation:**



This symbol can be used to illustrate the movement of entities/items from one physical location to another during processing.

**Boundaries:**



This symbol may be used to represent where the process or activity relates to another process or activity.

#### 5.4.3.2 Process Documentation Guidelines

Process models are the graphical documentation for how a process is supposed to work. However, the graphical documentation is limited in its ability to describe detail. Deliverables such as Process Descriptions must supplement the model with the other information required to fully describe the process and how its operation. These textual deliverables can include additional data, assumptions, and business rules need to complete the model and validate its accuracy.

Information on the following items should be collected and related directly to the elements of the process model:

Item Type	Data to be Collected	Other Tips and/or Related Data to Collect
Description	One or two sentence description of step	Precise but brief so that reading the statement brings an immediate recognition of the step, activity, or task.
Resources Required	Type and number of resource roles required to process one entity at this step	Describe the level of personnel and skill required for the process. The description should include any teamwork required or different resource types (e.g., senior, junior, trainee) and their priorities of involvement at this step. Include non-HR requirements (e.g., equipment)
Hands-on Processing Time	(Minimum, Maximum, Most of time) is preferred; (Minimum, Maximum) if "most of time is not relevant."	SMEs usually provide clear and accurate information if good facilitation and validation occurs Do NOT include estimated queue time, or travel time between steps. Document conditions causing Minimum and Maximum hands-on processing times - these conditions identify opportunities for improvement.
Business Rules and Policies	Decision criteria that dictate differences in processing approach and/or next process step	Be complete (i.e., exhaustive) and clear; otherwise the process model may not be accurate
Rework	Percent of time that result of this activity is incomplete or in error. Previous step (or this step) to which the entity must be sent for completion or correction.	Identify causes of incomplete and/or incorrect entities at this step.
Transfer/ travel to next step	Mode of transfer. Average time to transfer.	If transfer time is relatively large, collect Minimum, Maximum and Most of the Time data and document conditions for Minimum and Maximum times.
<b>Entity(ies)</b>		
Type	Name and short description	Separate entities by major groups such as retail vs. wholesale customer, etc. Use attributes to describe other differences within major groups.
Attributes	Relevant characteristics used by business rules to	For all attributes used in routing/processing decisions, list each possible attribute value

Item Type	Data to be Collected	Other Tips and/or Related Data to Collect
	determine routes.	
Quantity	Number of entities arriving into the process per time period	Obtain several periods (preferably one to two years) of recent historical data. Analyze patterns and variability in volume due to seasonality, product mix, etc. For “To-Be” process models, determine if volumes and quantities are likely to change, and investigate reliability of forecasts.
<b>Resources</b>		
HR Role (1 to n)	Description of all responsibilities in this process and other processes. Competencies and skills (actual and desired)	Describe any membership in teams with other HR types related to the processing of entities. Note perceived/ expected differences in relative efficiencies between teams of differing HR types.
Information System Application (1 to n)	Description of all uses in this process and other processes.	Note information regarding how IS applications automate decisions, processing flows, and the required data to support the process.
Other Resources (1 to n)	Description of all responsibilities in this process and other processes.	Note other resources required to complete activities and tasks within the process model. This can include job aids, reference cards, and other elements of the work environment.

The documentation details the process model and is relevant for both “As-Is” and “To-Be” processes. In cases where actual data is not available, assumptions must be made. This is especially true for “To-Be” processes. Documented assumptions are preferred over ignoring a portion of the documentation due to a lack of actual data. Most, if not all, of this information should be collected during group facilitation sessions

In addition, other general items of information should be documented. The nature of the sponsoring organization and the project will determine the amount and nature of this information:

- Key performance metrics, past achievement results, future changes in targets, etc.
- Cost information (fixed, variable, cost drivers, activity-based costs)
- Time information (total and by entity-type cycle times, current and future expectations)

#### 5.4.3.3 Process Levels and Decomposition

Processes can be modeled at different levels, depending on the objectives of the project and the sponsoring organization. In addition, high-level process models can be “decomposed” to lower levels at later stages in a project if necessary. A consistent set of guidelines and terminology is should be used when modeling processes at different levels.

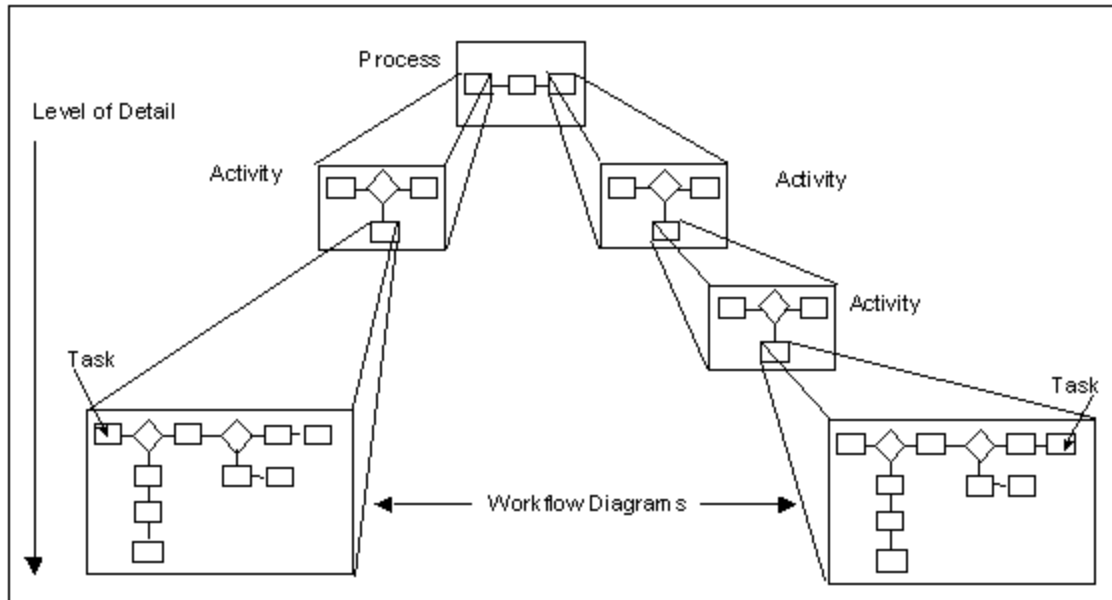


Figure 2 Example of Process Levels

Figure 2 illustrates how:

- Processes are decomposed into many activities
- There can be one or multiple levels of Activities
- Activities are decomposed into tasks
- Tasks are represented sequentially in a Workflow diagram

Examples to illustrate Process levels include the following:

Level:	Example:
Process	Fulfill Demand, Develop Products, Generate Demand, Plan and Manage Enterprise, Fulfill Order, Create Market, Manage Customer Relationships
Activity	Produce Stock Inventory, Develop Customer Strategies, Build Customer Relationships, Monitor Customer Needs
Task	Record Customer Details, Check Credit, Approve Order

### Why Decompose Processes?

From a simple viewpoint, a “top level” Process (such as Generate Demand) is made up of the sum of the lowest level tasks (Place Order). In many complex businesses, however, describing a Process simply in terms of all of its tasks is unwieldy and difficult to understand. Some organizations do adopt this freeform approach, illustrated by very detailed workflows. Without decomposition, such large workflows are complex, difficult to interpret, communicate, and combine.

Decomposition requires discipline and rigor, but delivers significant benefits. Decomposition is also enabled by a framework, (e.g. standards) to provide consistency. The benefits obtained from decomposition include the following:



- Aids understanding and interpretation
- Aids communication
- Increases manageability of process models
- Easy to combine across Processes

#### 5.4.3.4 Decomposition Approaches

##### **Functional Approach**

The functional approach to decomposition breaks down functional areas by decomposing until a level of detail is reached that represents a single unit of work. Each level of detail is independent of the level above. The example below, Aircraft Maintenance, illustrates the decomposition of the process into lower level functions and units of work.

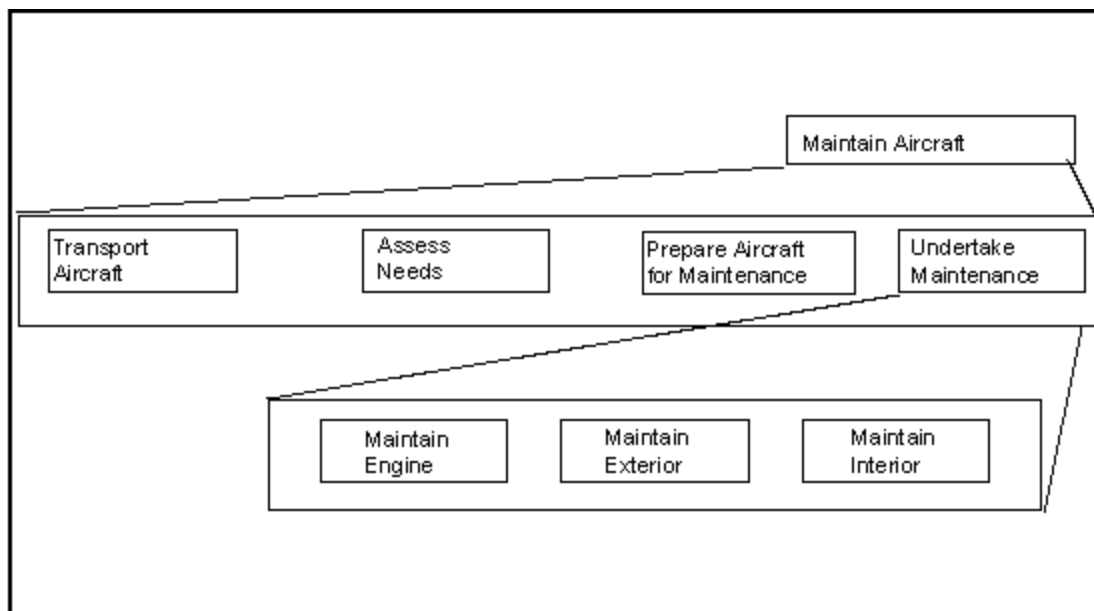


Figure 3. Functional Approach

##### **Strengths:**

- Practitioners are familiar with, and skilled in, traditional functional modeling approaches
- Tools which support this approach are well developed
- The process models can be expanded or broken down to accommodate different levels of complexity (Models are scaleable).

##### **Weaknesses:**

- Difficult to capture complexity in business processes
- Opportunities for re-use is limited as information becomes specialized and specific as it is decomposed

## Characteristics of Levels of Processes

An explanation of the characteristics of each level of the process model is described below.

<b>Process</b>	<b>Activity</b>	<b>Task</b>
Process Owner has responsibility for process	Potential for one individuals to be responsible for multiple activity	Business representative responsible for completion of task
Owner does not execute but manages/has ultimate responsibility	Owners participate/execute	Business Representative undertakes task
Process crosses Organization boundaries	Activity can be in one organizational unit or could cut across organization boundaries	Within one organizational unit
Processes are cross competency and multi- functional in nature	Single competency, but can still be multi-functional	Single competency and uni-functional
Process incorporates strategic objectives	Activity objectives derived from the strategic objectives	Often not reported. Can be measured in terms of response times or as part of the higher level performance measure

### Top Down & Bottom Up Approach

Process decomposition breaks processes into activities and tasks. Tasks can be considered to the lowest level of process decomposition because further decomposition is impossible or unnecessary. Further decomposition may be unnecessary if the implications of the business process are understood in sufficient detail for explanation purposes. If there is application and job development occurring as a result of the business process design, the tasks would represent the lowest possible unit of work level. This approach incorporates a top-down view of processes.

To follow a bottom-up view (i.e. upward amalgamation) of Processes, the relationship or communality between the process, activities, and tasks can be examined. These relationships can be categorized by:

- Common inputs and/or triggers
- Common outputs and/or outcomes
- Value
- Quality
- Volumes
- Assumptions
- Costs
- Time (elapsed and hands-on)
- Metrics
- Resources

All of the above categories can be incorporated into a higher level or broken down.

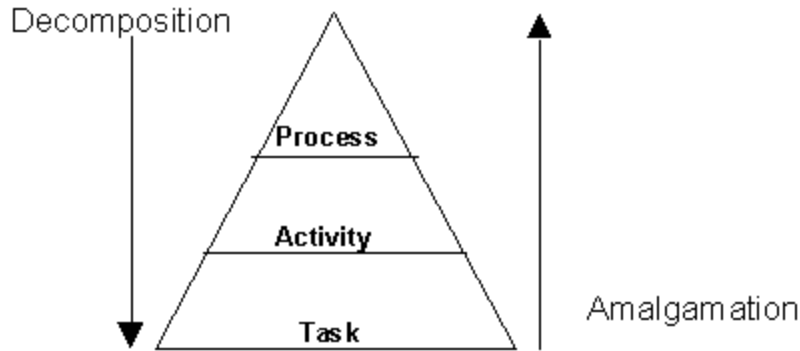


Figure 4. Top Down and Bottom Up Approach

A performance metric is an example of a category that can be drilled down and up. The highest level metric, for example, orders should be processed and delivered in 24 hours. This can be broken down in terms of detail, for example, Produce Stock Inventory in 1 hour. This can then be decomposed again, Enter Order in 2 minutes etc. The lowest level should be able to be amalgamated up to the highest level metric.

#### 5.4.3.5 Current vs. “To-Be” Process Levels

The objectives/reasons for modeling current business processes differ from those for modeling the “To-Be” process. As a result, the level of detail that may be required differs also. If the project is undertaking streamlining, gaining a detailed understanding of the current process becomes more important; whereas if radical re-engineering or a “blank paper” approach is being adopted, the need to understand current processes is minimal.

The current process assessment is often used for demonstrating an understanding of the business in its current state. It is used also to:

- Identify opportunities for improvement.
- Identify the gap between the sponsoring organization’s business needs and requirements and where they are now.
- Gain and demonstrate an understanding of current operations and performance.
- Identify further requirements for the process.
- Understand the constraints the sponsoring organization faces.
- Identify parts of the processes, activities, and tasks that add value.
- Identify parts of the current process that are non-value added (in the eyes of the Customer, Suppliers and the Sponsoring Organization).
- Identify weaknesses in the current performance measurement system.

When designing the “To-Be” processes, activities, and tasks, there are no fixed rules as to the level of detail required. This often depends on the situation; level of detail differs depending upon the organization’s needs, approach, and the scope of the capability release. There are however, some

general questions that may help in deciding, for each process, how far to go in terms of detail. The importance of detail is likely to increase if the answers to the following questions are yes.

There are some general questions that can be asked to help identify the level of detail required when modeling the current workflow diagrams. If the answer to many of these questions is yes, it may be worth drilling down into more detail.

- Is this process a priority process?
- Does this process represent the identity of the organization and form customer perceptions?
- Is this process highly complex?
- Is the process subject to many regulations (e.g. safety regulations, ethical considerations, etc.)?
- Is there a high degree of risk associated with the process (e.g. aircraft maintenance)?
- Is there a high degree of customer interaction?
- Does the process have considerable potential for efficiency savings?
- Can the process contribute to increasing competitiveness?
- Does the process greatly contribute to profitability?
- Is this process new, as opposed to an existing process that has been redesigned?

More specific questions can be asked, in order to help identify if it is logically possible to decompose further. This will help answer the question of when to stop decomposing and provide “exit criteria.” This again interacts with the scope and approach of the specific capability release. Questions to refer to in deciding when to stop include:

- Could the next level of processing be classified at the level of a user procedure (automated)?
- Is the next level at the discretion of the business representative? (e.g. In the process “Resolve Customer Complaint,” the task “appease customer” may not be decomposed to avoid being prescriptive, and may instead rely on the business representative’s discretion and experience.)
- Will the next level of detail add value in terms of understanding the processes?
- Can the process be implemented with this level of detail or will further decomposition and detail be required to make the process implementable?

#### **5.4.4 Swimlane Diagram Guidelines**

Swimlane Diagrams provide a method for illustrating a detailed interaction model for a Business Capability. The diagrams are driven by the Business Process Designs, and they extend the detail-level process model (or a workflow diagram) into a model that depicts the detailed interactions of the Business Process with other elements of the capability and identifies implementation implications for capability. The documentation for the process model should already contain this information and the Swimlane Diagram converts that part of the documentation into a graphical format.

By extending a detailed-level process model (or workflow diagram) into a Swimlane Diagram format, it may be easier to identify and emphasize implications for:

- Applications and application requirements needed to support the Business Capability
- Physical Environment requirements needed to support the Business Capability
- Performance Support requirements to support the Business Capability
- Technology Infrastructure required to support the Business Capability

Consider the following example:

A Workflow diagram is produced to depict the business process flow:

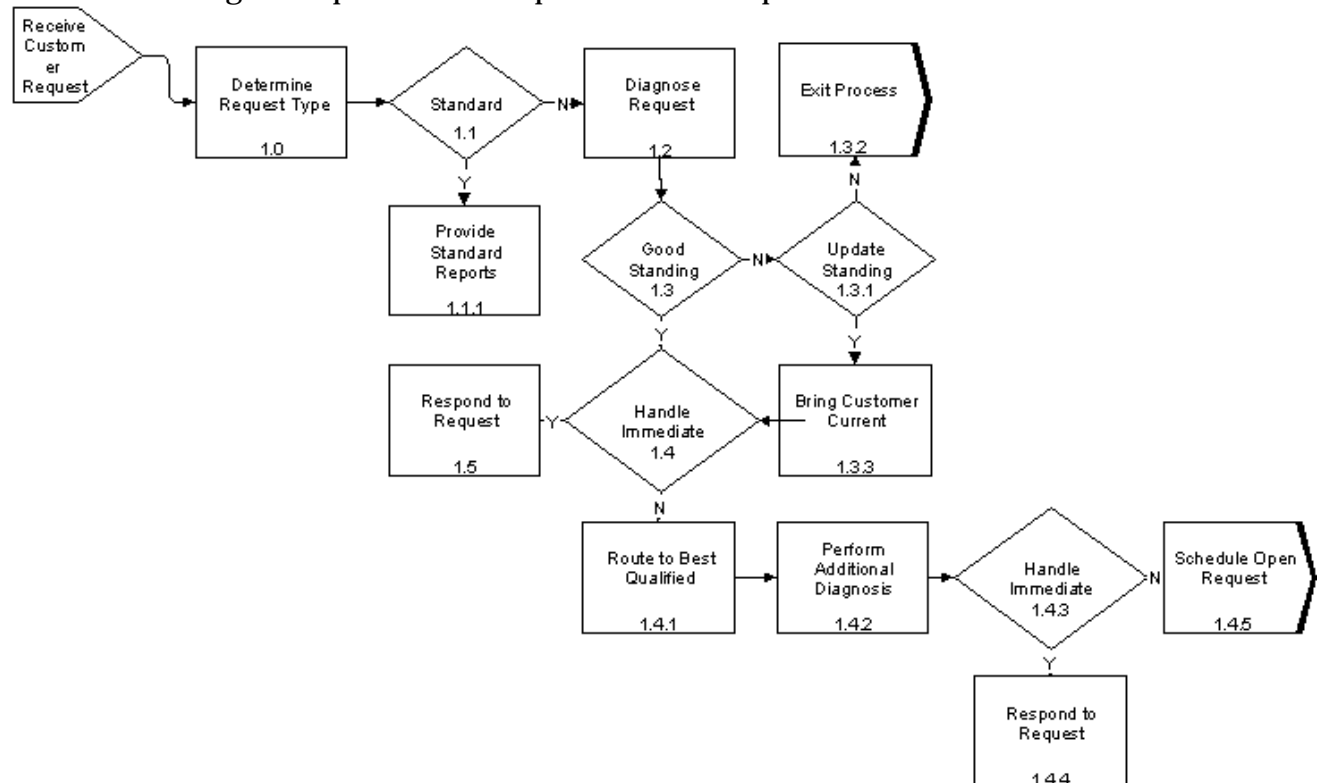


Figure 5. Workflow Diagram

The Workflow Diagram helps the business users and the design team to understand the flow of work tasks within the business process, however information on resources required (human and application) to execute the tasks, where the tasks are executed, and what areas of the process will require performance support is contained in the supporting Process Documentation.

By re-drawing the workflow diagram into a Swimlane Diagram (see Figure 6), the Business Process flow is still understood but supporting information is provided pictorially so that the implementation implications for the Business Capability may be better understood. Synergy for tasks are identified. All the tasks within one lane share the attributes of the driver (or entity) for that lanes (i.e. role).

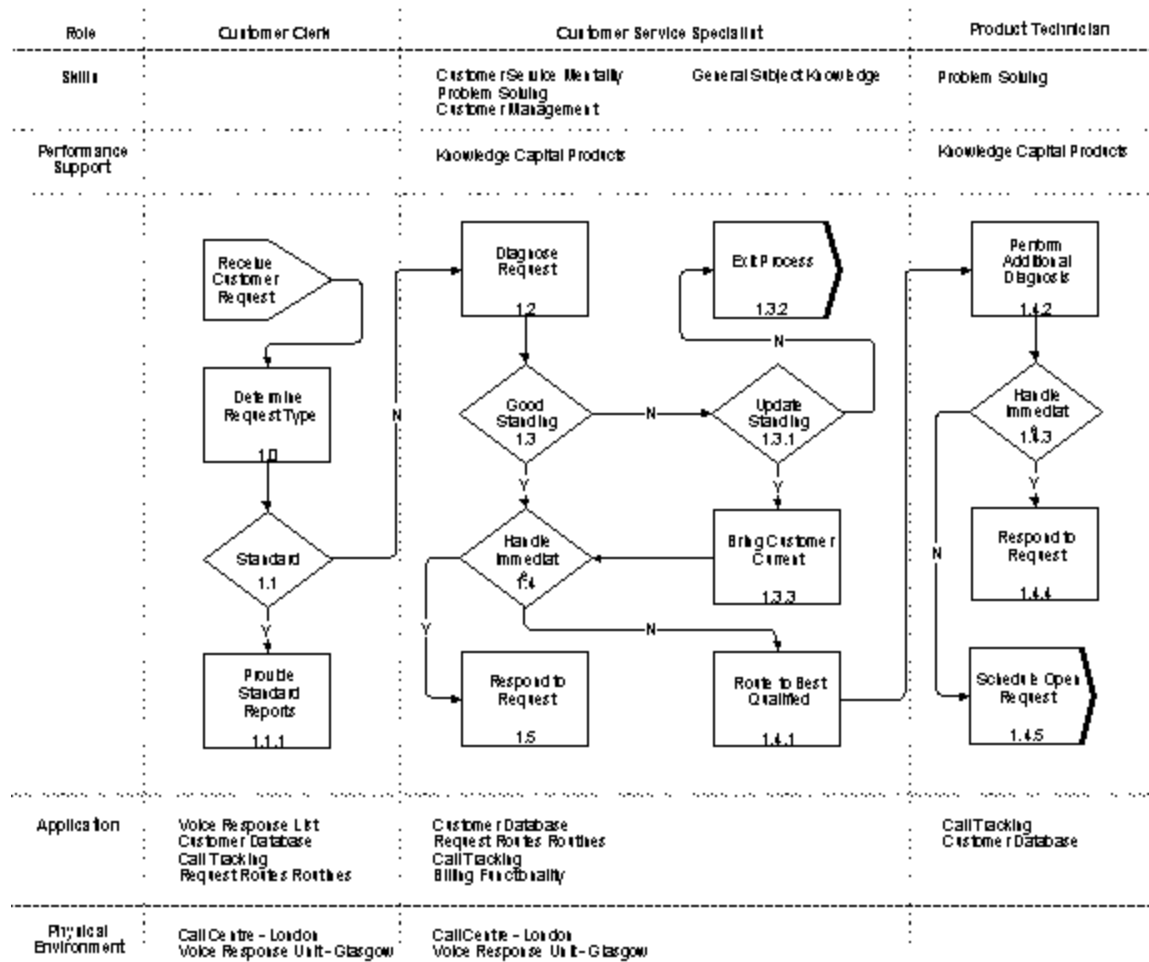


Figure 6. Swimlane Diagram

The Swimlane Diagram provides detailed information which can help to inform design teams of the sequence that the tasks within a process are executed, the resources required to execute the task, performance support required to execute the task, and the physical environment required to execute each task. The information is produced in an integrated fashion (the Swimlane Diagram) to show the ‘bigger picture’ to design teams, but effectively from this diagram, detailed information can be derived for the design teams, for example:

Interaction	Role			Application					Physical Environment		Skills				Perf. Support	
Tasks	Customer Clerk	Customer Service Specialist	Product Technician	Voice Response List	Customer Database	Call Tracking	Request Routes Routines	Billing Functionality	Call Centre - London	Voice Response Unit - Glasgow	Customer Service Mentality	Problem Solving	General Subject Knowledge	Customer Management	Yes (Knowledge Capital Products)	No
Receive Customer Request	X			X	X	X	X		X	X						X
Determine Request Type	X			X	X	X	X		X	X						X
Standard	X			X	X	X	X		X	X						X
Provide Standard Reports	X			X	X	X	X		X	X						X
Diagnose Request		X			X	X	X	X			X	X	X	X	X	
Good Standing		X			X	X	X	X			X	X	X	X	X	
Handle Immediate		X			X	X	X	X			X	X	X	X	X	
Schedule Open Request			X		X	X						X			X	

Figure 7. Objectives of a Swimlane Diagram

#### 5.4.4.1 Objectives of a Swimlane Diagram

As indicated in Figure 7, the objectives of the Workflow Diagram or process model are to illustrate the flow of work in the new Business Processes. The Swimlane Diagram extends the process flow to emphasize interactions of all elements of the Business Capability. Its main objectives are:

- To illustrate the interactions between the Business Process and the application, users, Physical Environment and Technology Infrastructure.
- To show an integrated view of the Business Capability.
- To illustrate the implementation implications of the new Business Process Designs.
- To provide a starting point for further design of the Application, Human Performance and Technology Infrastructure elements of the Business Capability.

#### 5.4.4.2 Current vs. “To-Be” Swimlane Diagrams

Swimlane Diagrams may be created when analyzing a Current Business Capability or when designing a new Business Capability. The information contained in the Swimlane Diagram is very detailed (task level) so it may not always be created for current analysis. This will depend on the cost benefit of doing the detailed analysis.

Usually, the expected approach for creating a Current Swimlane Diagram are different from those for creating a “To-Be” Swimlane Diagram.

When creating a Swimlane diagram to depict a current Business Capability, consider:

- Where are the hand-offs in the Business Process?
- How does the Business Process operate in terms of cost, quality and time?
- How does the current capability interact?
- Which tasks are critical?





will depend on the view of the Business Process that is required. There can be multiple views of the same workflow, each focusing on a different driver.

- Sequentially model each task into the appropriate lane (ensuring that sequencing and linkages between tasks remain exactly the same as in the process models or Workflow Diagrams). For example, each task from the Workflow Diagram (Figure 5) is placed into the appropriate lane. (Figure 9)

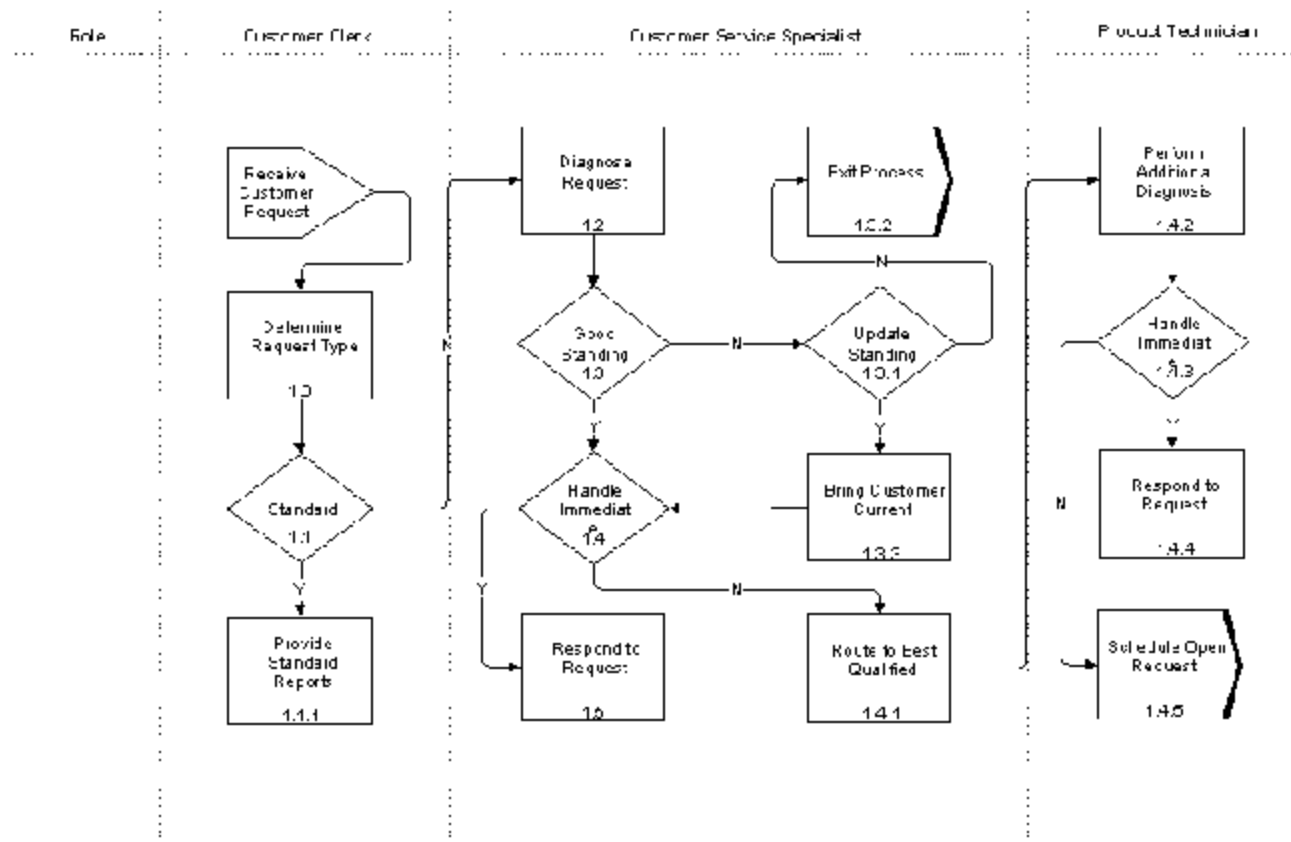


Figure 9. Lane Assignment.

- Add additional lanes to depict required supporting information. For example, the Swimlane Diagram in Figure 10 has lanes related to Skills, Performance Support, Application Interaction and Physical Environment added to provide supporting information and provide design teams with a starting point for design.

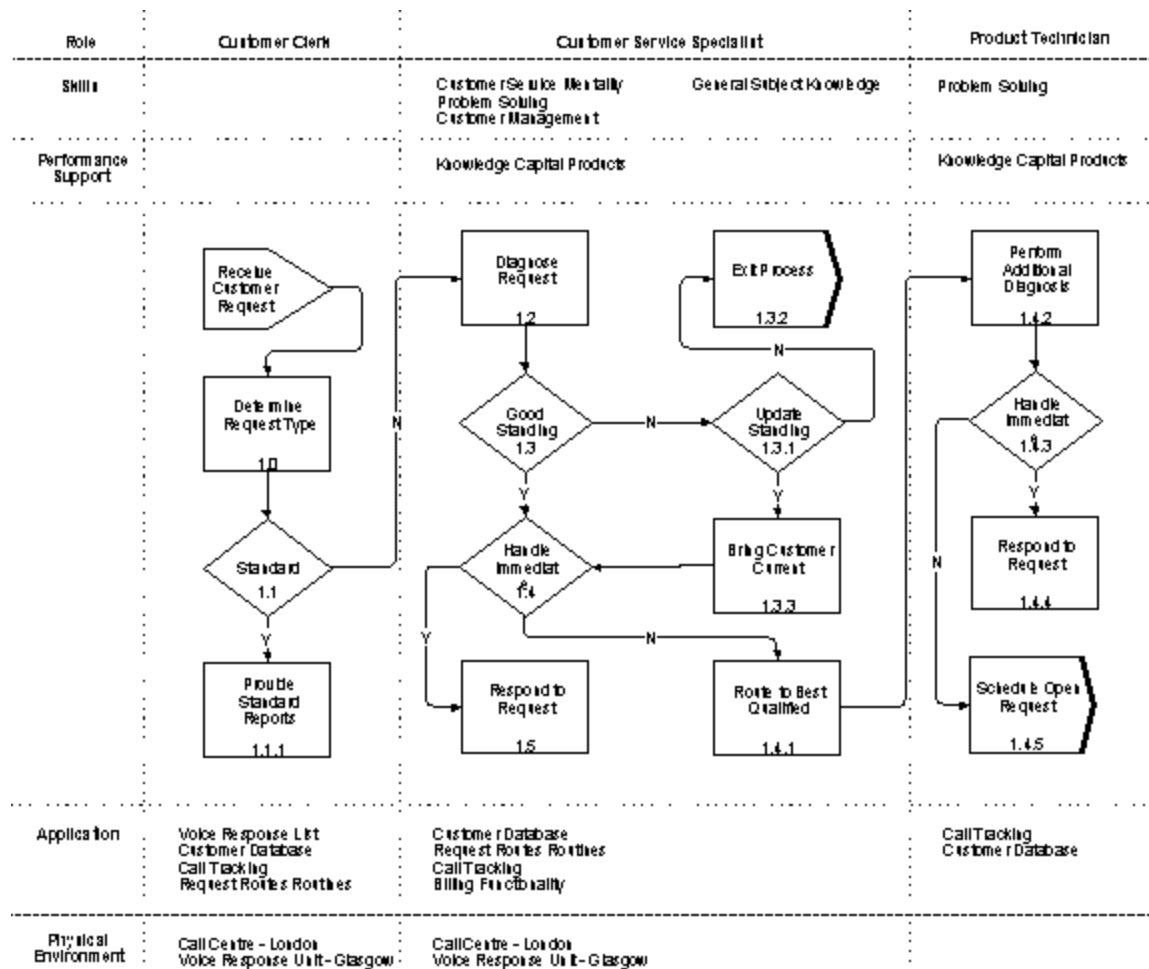
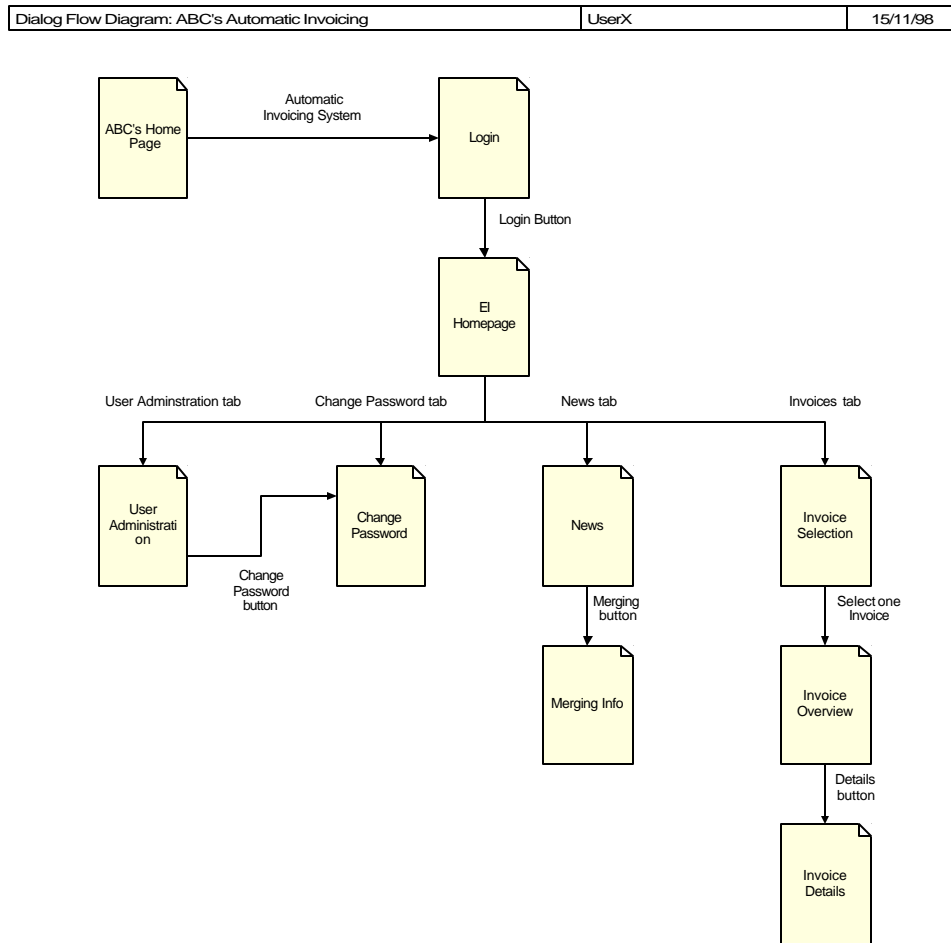


Figure 10. Swimlane Diagram

- Validate the Swimlane Diagram against the process model or Workflow Diagram (and other supporting information from the Business Process Design) to ensure that the flow of tasks and dependencies remain stable.

## 5.5 Sample Dialog Flow



### Sample Description

An application can be made up of more than one dialog, so each application may require several dialog flows to explain it. The diagram is useful because both graphical user interface (GUI) and web-based applications are often made up of many small, interconnected windows and/or pages. A dialog flow diagram can simplify the presentation of the user interface when explaining it to management or users. The two sample dialog flow diagrams (one for a GUI application and one for a web-based application) illustrate the following components:

- **Windows/Web Pages.** The windows and pages that make up the dialog are depicted by labeled rectangles. “List of Customers” is a window. “ABC’s Home Page” is a page.
- **Arrow.** Arrows indicate the window/page created when the user takes some action. Note that an arrow can connect a window to itself, for example, when an “Add” push button updates something on its own window but does not pass control to another window.
- **Arrow Label.** Each arrow is labeled with an identification of the widget(s) and actions that transfer control from one window to another. For example, OK Push Button is an example of such a widget (for push buttons, the action “click” is understood since that is

the only action allowed). If the response to the widget action pairing is conditional (different windows will be open in different conditions), the required condition is placed in parentheses. For web-based applications, the arrow labels indicate the action required to navigate from one page to another.

- **Common Windows.** Some Windows/Web Pages are available from most or all of the windows in the application (for example, a help window/page). Instead of showing the same window/page multiple times on the diagram, it is more efficient to create a note box off to the side.

## 5.6 Scenario Development Technique

### 5.6.1 Workflow Scenarios

#### 5.6.1.1 Guidelines

You may want to document scenarios at two levels of detail: the business process level and the business activity level. For a business process, each scenario should describe the implementation of a business practice under specified conditions. For a business activity, each scenario should describe the implementation of the business activity under specified conditions in the context of a business practice.

It should be possible to trace all of the scenarios for a business process through the workflow diagram(s) for that business process. The same should hold true at the business activity level.

Determining the right number of scenarios is more of an art than a science, but generally speaking more variability will require more scenarios. The sum of all scenarios within a business process or activity should provide a representative sample of the variability that must be supported within the business or business activity. You don't need a scenario for every possible path through the business process or activity, but you will benefit from focusing on the most common and most important scenarios. Create a variability matrix to determine the most common and most important sources of variability (see the Using a Variability Matrix to Document Workflow Scenarios guideline and the Workflow Variability Matrix deliverable).

You may want to rate scenarios, and even scenario steps, from a variability perspective. For example, try to answer some of the following questions:

- How frequently is this scenario (or scenario step) performed?
- What percentage of scenarios follow a path similar to this one?
- How important is this scenario (or scenario step)?

It is important to ensure consistency within and across scenarios, as well as consistency between workflows and scenarios.

To maximize the effectiveness of this deliverable, use language that can be understood by a broad audience: business experts and users, user interaction designers, and training team members. Using active verbs throughout this document will help keep the focus on actions rather than results.

Be sure to cover a broad range of business process requirements.

Techniques:

- Use a variability matrix to identify sources of variability (see the Using a Variability Matrix to Document Workflow Scenarios guideline and the Workflow Variability Matrix deliverable). Another option is to conduct a brainstorming meeting on variability.
- Create scenarios to illustrate sources of variability. Start with the most common and most important.
- Use scenario descriptions from business practices as a starting point.

## Volume

Many per solution

### 5.6.1.2 Sample Workflow Scenario

Business Practice: [1] XXXXXXXXXX

Scenario Description [2]

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

#### Scenario Steps

# [3]	Step Name [4]	Actor [5]	Triggers [6] (Events and Rules)	Data Inputs (Information) [7]	Step Description [8]	Data Outputs [9] (Information)
1.						

## Sample Description

This sample deliverable is at the business process level. It contains the following information:

**[1] Business Practice**---The business practice for which the scenario is written.

**[2] Description**--- A 1-2 sentence description of what happens during this scenario.

**[3] Step**--- The scenario step number.

**[4] Step Name**---A brief description that captures the essence of the scenario step.

**[5] Actor**---Answers the question: Who performs this step?

**[6] Triggers**---The events and/or business rules that cause this step to happen. Explains what causes the actor to perform this step.

**[7] Data Inputs**---The data inputs required to perform this step.

**[8] Step Description**---Describes what happens within this step.

**[9] Data Outputs**---The data outputs produced by this step.

You may also decide to gather the following additional information for each scenario step (not shown in the sample):

### 5.6.2 Scenario Gathering

This section describes how to gather detailed scenarios by using the concrete knowledge of the business representatives, other stakeholders, and content experts. Use the scenarios, when developed to verify and validate Business Process Designs. The scenarios can also potentially be used for the validation of the entire capability release. Gather scenario data with this in mind.

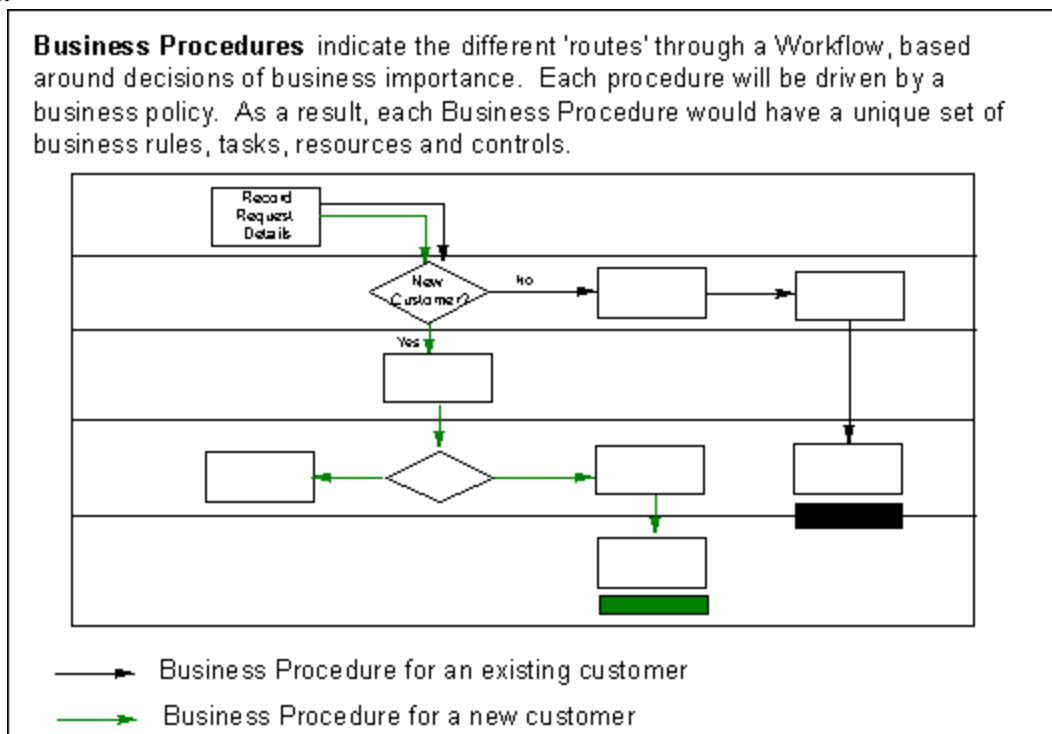
To use this technique, you must have a complete understanding of the business processes that are in scope. Ensure the team understands the relationship between Business Processes and Business Procedures depicted below.

#### Process

A set of logically related and continuously evolving business activities, or lower level processes, that combine to satisfy a business objective.

#### Business Procedures

Business Procedures indicate the different “routes” through a Business Process, based around decisions of business importance. Each procedure will be driven by a business policy. As a result, each Business Procedure would have a unique set of business rules, tasks, metrics, resources, and controls.



You will need the following information:

- A description of each business process, including a breakdown of its activities and tasks
- A description of how applying a business procedure affects business processes

The Business Process Design and the Business Procedures Inventory are the two primary sources for this information. In addition, your input should include a scenario coverage plan that provides structure to the gathering process.

### **5.6.3 Scenario Gathering Process**

#### **5.6.3.1 Interview Sources**

To begin each interview, explain the purpose of the interview. Describe scenarios and how you will use them. Refer to an actual scenario in an unrelated area. Then document any concrete contextual information that “sets the stage” for the scenario. For example, document:

- The business procedure being followed.
- A short description for the scenario.
- Any performance requirements or other special requirements.

Use a simple method to capture a scenario. For example, ask the interviewee to describe in detail the steps that comprise a business process, and capture this information on a blank page, on a whiteboard, and/or using a recording device. Although you may use a freeform technique, keep in mind that you’re seeking specific information for the scenario deliverable:

- The actor who performs the step.

**Note:** If you first identify all the actors and then derive the scenarios from them, be sure your list of actors includes people you might not traditionally specify (e.g., operations people).

- The events and rules that trigger the step (control inputs)---To elicit this information, ask questions like the following: “How do you know when to perform this step?”
- The information required to complete the step (data inputs).
- The conversions that define what happens as part of the step---You are also interested in business rules that affect the step.

**Note:** At each step of the scenario, you may identify the exceptions that could occur, but do not follow the exceptions through their branches (e.g., the Workflow Diagram describes what happens when those exceptions occur). For example, a customer does not have any outstanding payments that are over 60 days past due, so the application approves credit. For this scenario, you would also document, at a high level, what happens if the customer does have past-due payments. You would not, however, continue describing the past-due exception path.

- The events that result from the step (control outputs)
- The information created or updated by the step (data outputs)



- Performance requirements for the step. For example, how long should a step take? How much time should pass in between scenario steps? How is it indicated that the step has been performed well? Poorly?

**Note:** The Performance Support Requirements defined in the Identify Performance Support Requirements task may provide this input.

Finally, keep in mind that it is the responsibility of the gatherers to collect scenarios that:

- Pass through their business processes.
- Reflect the business procedures that are in scope for their business processes.
- Reflect the variability across and within the business procedures.

#### **5.6.3.2 Document Scenarios**

Document each scenario by referring to your paper notes, whiteboard notes, audiotapes, and/or videotapes. As you perform this step, you will likely encounter missing information. Keep a list of questions to be asked of your sources. Contact them at a later date to fill in the missing information.

#### **Evaluate Coverage**

Determine how well your scenarios cover the business processes and procedures that are in scope. Are there significant gaps? Do your scenarios cover the variability across and within each business procedure? If gaps exist, repeat the process until you have covered them.

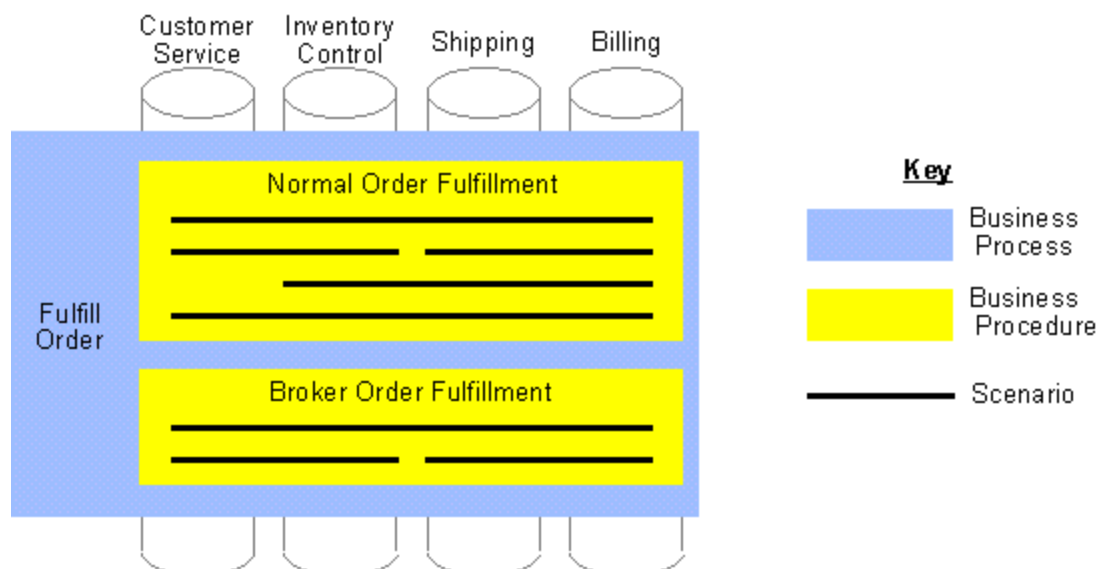
### **5.6.4 Tips and Traps**

#### **5.6.4.1 Scope**

Throughout the scenario gathering process, be sure you understand the scope of the project. Do not waste your time charting territory that is outside scope boundaries.

#### **5.6.4.2 Business Procedures**

Collect a separate set of scenarios for each business procedure within a business process. For example, if the Fulfill Order business process is in scope and if the Normal Order Fulfillment and Broker Order Fulfillment business procedures are in scope, collect two complete sets of scenarios -- one for each business procedure (Figure 1).



**Figure 1:** Two sets of scenarios -- one for each business procedure

#### 5.6.4.3 Types of Business Tasks

As you document scenario steps, be sure to document all types of business tasks (and/or activities). By doing so, your scenarios will more effectively uncover process improvement opportunities. Business tasks (and activities) may be categorized using two dimensions. The first dimension distinguishes between computer-enabled and manual tasks, and the second distinguishes between business representative and non-business representative tasks. When you plot these dimensions, four categories reveal themselves.

	Computer-Enabled	Manual
User		
Non-User		

- A computer-enabled, business representative task involves a person interacting with a computer. Another name for this type of task is “semi-automated.” Example: an order entry clerk typing a customer order into a computer.
- A computer-enabled, non-business representative task involves only a computer. Another name for this type of task is “automated.” Example: a computer accepting an EDI customer order.

- A manual, business representative task involves only a person. Another name for this type of task is “manual.” Example: an order entry clerk deciphering a code using a “cheat sheet” tacked to his or her bulletin board.
- A manual, non-business representative task involves neither a person nor a computer. Instead, it usually involves a piece of equipment (other than a computer) that provides some degree of automation. Another name for this type of task is “automated.” Example: a conveyor belt delivering goods to the shipping dock.

#### 5.6.4.4 More Tips and Traps

- A scenario should not include alternative branching. It should represent a single-threaded path through a business process.
- Use specific names for actions in the scenario. This encourages you to write a more concrete scenario. For example, “Sally enters the order information,” not “The order entry clerk enters . . . “
- Capture the frequency and importance of each scenario. Also, quantify scenarios based on project objectives such as cost, cycle time, transaction volume, etc.
- Develop scenarios with testing in mind because the scenarios will drive the creation of product test conditions.
- Write scenarios using active (rather than passive) language. For example:

Active: Enter the order information.

Passive: The order information is entered.

**Note that in the passive-language example above, you cannot tell who performs the action. The active language example states the business representative (the implied “you”) and is more concise.**

## **5.7 Using a Variability Matrix**

This guidelines describe a technique for creating a Workflow Variability Matrix. This process helps you to identify the drivers or sources of business process variability, and to use this information to determine how many and which scenarios to document. The scenarios are used to verify and validate business process designs.

Use this guideline to ensure that all types of business events and other significant conditions are covered by the defined business processes. This guideline is particularly useful for domains with complex or highly variable business processes. Use it with section 5.6 Scenario Development Technique.

### **5.7.1 Overview**

The objective of creating a Workflow Variability Matrix is to brainstorm and summarize the factors that cause the business processes to vary and to identify scenarios that fully describe the scope of the solution.

The ultimate objective is to make sure that the scenarios and workflow diagrams are complete and account for the full breadth and depth of business process variability. The technique you use to accomplish this objective is not as important. The technique described in this guideline, which is basically a structured brainstorming technique, is only one of many possibilities. It concentrates first on brainstorming all possible sources of variability, and then on identifying scenarios to illustrate those sources of variability.

Each scenario should document one path through a business process. Together, all of the scenarios for a business process should span the variability that is in scope. Scenarios may apply at any level of the business process model. For example, information at the process, activity, or task level could be validated. See the Process Modeling guideline for more information about the levels of a business process model.

It is important to perform this session with business experts in the area discussed. However, it is also extremely important that the scope is solid. If there are any questions about scope, return to the business process design and resolve any open issues. If it is not appropriate to resolve the open issues in a large group, move the discussion to a smaller group. Ensure that the key stakeholders and business experts are present to make the appropriate decisions.

### **5.7.2 Facilitation Process**

The facilitation technique described here will make the process of discovering variability and identifying scenarios easier. The following steps are outlined below:

1. Prepare the conference room
2. Introduce the facilitation process
3. Determine how many and which scenarios to document
4. Document the baseline scenario in a large group
5. Document the remaining scenarios in small teams

## 6. Wrap up session

### 1. Prepare the conference room

Typically, at least one large white board is required. It is also useful to have secondary whiteboards to document further information (e.g. open issues, points to address later.) Ensure that you have all necessary supplies on hand.

### 2. Introduce the facilitation process

Explain the facilitation process to the participants. Because it is not very complicated, you may explain it without using a presentation. Use the following outline to guide your explanation:

Identify drivers or sources of variability and determine which scenarios to document

Document the baseline scenario is a large group

Document the remaining scenarios is small teams

If it is appropriate for your group, refer them to the Scenario Planning Technique and Scenario Gathering Technique guidelines.

### 3. Determine how many and which scenarios to document

The purpose of this step is to identify the variability using the Workflow Variability Matrix and then use this information to determine how many and which scenarios to document. Unless your scope is narrow and well-defined, it is usually difficult to make this decision. You could spend too much time collecting scenarios, so it is very important to gather only enough scenarios to fully describe the variability that must be supported by the business process design.

The following description represents a complete approach for using the Workflow Variability Matrix, but you may refine this approach to meet your specific needs.

The Business Capability Requirements serve as the primary input into this step. Use the Business Capability Requirements to understand the amount of variability that must be supported by the solution and the various paths through the business process that need to be validated. Use a subject area data model to relate that variability to characteristics of specific business entities.

Hand out these deliverables and ask the participants to review them for five minutes. Draw an empty Workflow Variability Matrix on the white board while the participants are reviewing the process designs.

When the group is ready, begin brainstorming by asking the following questions:

What is the business process/activity/practice you are trying to enable?

What is the primary business entity that participates in the business process/activity/practice?

In what ways do the characteristics (i.e., possible states) of this business entity affect the workflow?

What other business entities participate in this business process/activity/practice?

In what ways do the characteristics (i.e., possible states) of these business entities affect the workflow?

Do different business events result in different workflows?  
What situations or conditions cause a change to the workflow?

As the participants provide answers to these questions, build the Workflow Variability Matrix. Here is an example to illustrate the process. Before you have asked any questions, the Workflow Variability Matrix should look like:

		V ariability in Secondary Business Entity											
V ariability in Primary Business Entity	<u>Business Process</u>	Scenarios											
	<u>Primary Business Entity</u>												

- 1) What is the business process/activity/practice that you are trying to enable? **Fulfill Order**
- 2) What is the primary business entity that participates in the business process/activity/practice? **Order**

		V ariability in Secondary Business Entity											
V ariability in Primary Business Entity	<u>Business Process</u>	Scenarios											
	<i>Fulfill Order</i>												
	<u>Primary Business Entity</u>												
	<i>Order</i>												

- 3) In what ways do the characteristics (i.e., possible states) of this business entity affect the work flow? **Consignment orders are processed differently than standard mass merchandise orders.**

**And even standard mass merchandise orders are processed differently if the order value is greater than \$1,000.**

The Workflow Variability Matrix should now look like this:

		V ariability in Secondary Business Entity											
V ariability in Primary Business Entity	<u>Business Process</u> <i>Fulfill Order</i> <u>Primary Business Entity</u> <i>Order</i>	Scenarios											
	<i>Mass merchandise</i>												
	<i>- Order value</i>												
	<i>Consignment</i>												

4) What other business entities participate in this business process/activity/practice?  
**Customer, ...**

5) In what ways do the characteristics (i.e., possible states) of these business entities affect the work flow? **The customer's billing preference affects the work flow. For example, some prefer pre-billing. If a customer is a preferred customer, we ...**

The Workflow Variability Matrix should now look like this:

		V ariability in Secondary Business Entity											
V ariability in Primary Business Entity	<u>Business Process</u> <i>Fulfill Order</i> <u>Primary Business Entity</u> <i>Order</i>	Scenarios	<i>Customer</i>	<i>- Post-bill</i>	<i>- Pre-bill</i>	<i>- Preferred</i>	<i>- Not preferred</i>						
	<i>Mass merchandise</i>												
	<i>- Order value</i>												
	<i>Consignment</i>												

6) Do different business events result in different work flows? **Sometimes a customer calls to change an order. If the shipment has already been scheduled, we need to send a message directly to the shipping supervisor.**

7) What situations or conditions cause a change to the work flow? **If a customer orders a product that is not available, sometimes we can recommend product substitutes. But only for some products, and we always ask the customer if it is okay. If a product is not available in the primary warehouse, another option is to reserve inventory from an alternate warehouse.**

The Workflow Variability Matrix should now look like this:

Variability in Secondary Business Entity

Variability in Primary Business Entity	Business Process <i>Fulfill Order</i>	Scenarios	<i>Customer</i>	<i>- Post-bill</i>	<i>- Pre-bill</i>	<i>- Preferred</i>	<i>- Not preferred</i>	<i>Product</i>	<i>- No Substitute</i>	<i>- Substitute</i>	<i>Inventory</i>	<i>- Primary w/h</i>	<i>- Alternate w/h</i>
	Primary Business Entity <i>Order</i>												
	<i>Mass merchandise</i>												
	<i>- Order value</i>												
	<i>- Order maintenance</i>												
	<i>Consignment</i>												

As you can see, the Workflow Variability Matrix tends to grow very quickly. If you are planning to use this technique, keep in mind that the size of your scope will determine how much time you need. Protect yourself from attempting to accomplish more than is possible in a group setting. If the scope of your solution is too large to perform this step in a large group, you may want to perform it outside of the group setting. On the other hand, if your scope is narrow and well-defined, you may not need this step.

If you can arrive at the same information using a different technique, do not be afraid to try it. Remember that the objective of this step is to determine how many and which scenarios to document. After the group has identified the drivers or sources of business process variability, it is ready to determine how many and which scenarios to document.

It is best to start with the most common scenario, sometimes called the baseline scenario. Document the baseline scenario next to the most appropriate category of variability for the primary business entity. For this example, you would document the baseline scenario next to the “Standard Mass Merchandise.” Then work your way from left to right, considering each category of variability for the secondary objects. Ask the participants, “Should the baseline scenario include



this feature?" If the answer is "Yes," then check the corresponding box. After completing the baseline scenario for this example, the Workflow Variability Matrix would look like this:

Variability in Secondary Business Entity

Variability in Primary Business Entity	Business Process <i>Fulfill Order</i>	Scenarios	<i>Customer</i>	<i>- Post-bill</i>	<i>- Pre-bill</i>	<i>- Preferred</i>	<i>- Not preferred</i>	<i>Product</i>	<i>- No Substitute</i>	<i>- Substitute</i>	<i>Inventory</i>	<i>- Primary w/h</i>	<i>- Alternate w/h</i>
	Primary Business Entity <i>Order</i>												
	<i>Mass merchandise</i>	<i>Baseline scenario</i>		✓		✓			✓			✓	
	<i>- Order value</i>												
	<i>- Order maintenance</i>												
	<i>Consignment</i>												

Continue to propose additional scenarios until at least one check mark appears in each column and row that represents a category of variability. Describe each scenario well enough to provide guidance to the team that will be responsible for documenting it. After completing this step, the Workflow Variability Matrix would look like this:

Variability in Secondary Business Entity

Variability in Primary Business Entity	Business Process <i>Fulfill Order</i>	Scenarios	<i>Customer</i>	<i>- Post-bill</i>	<i>- Pre-bill</i>	<i>- Preferred</i>	<i>- Not preferred</i>	<i>Product</i>	<i>- No Substitute</i>	<i>- Substitute</i>	<i>Inventory</i>	<i>- Primary w/h</i>	<i>- Alternate w/h</i>
	Primary Business Entity <i>Order</i>												
	<i>Mass merchandise</i>	<i>Baseline scenario</i>		✓		✓			✓			✓	
		<i>Pre-bill, Product sub.</i>			✓					✓			
	<i>- Order value</i>	<i>Cust. not preferred, ...</i>		✓			✓						✓
	<i>- Order maintenance</i>	<i>Maintain order ...</i>				✓							
	<i>Consignment</i>	<i>Consignment order ...</i>		✓		✓							

You may want to document the Workflow Variability Matrix electronically as the group develops it. If you proceed immediately to the next step, you may need to erase the matrix, especially if your room is equipped with only one large whiteboard. It is also important to make the Workflow Variability Matrix available for the next step. It will serve as an excellent table of contents for your scenarios.

#### **4. Document the baseline scenario in a large group**

It is a good idea to document the first scenario in a large group. Start with the baseline scenario because it usually serves as an excellent reference for all of your other scenarios. The baseline scenario should cover the variability described in the Workflow Variability Matrix.

Start with a story. Ask one of the business domain experts to tell the story. The story should describe a realistic occurrence of the business process that is in scope. It should include real business events and business representatives with real names. Business data and business rules should be stated specifically. In fact, it is a good idea to specify actual values whenever the values might be important to the scenario. For example:

##### **Scenario**

Lucy, a call center agent, receives a phone call from a Pizza Place employee to notify ACME Insurance of a basic workers compensation claim for a cut finger. Lucy sets up a new claim and enters the name of the insured, the policy number, the state in which the incident occurred, and the date of loss. Lucy saves the claim. The application validates the claim information and automatically assigns the claim to Linus (as the owner) because he is the claim supervisor for the state in which the incident occurred.

Linus looks at his electronic in-box and invokes the task to Assign Claim Type. The application launches a user interface that allows him to assign a claim type. Linus assigns a claim type and notices in the task list that the new claim needs to be assigned to a claim handler. Linus selects Sally, an outside claim representative working out of her home, as the claim handler for the new claim. Linus marks the Assign to Claim Handler task complete.

Sally receives the claim in her electronic in-box and works on a couple of tasks, including the first contact for claimant. She uses some additional windows to capture claim data. A background process evaluates the available data and identifies a potential subrogation opportunity. (The claimant who was injured used a knife with a handle that broke, so ACME Insurance may be able to recover damages from the manufacturer of the potentially defective knife.)

The application adds the new task Alert the Recovery and Subrogation Unit to the suggested task list for the claim. Sally agrees with the suggestion and adds the suggested task to the planned task list. Because Charlie is the recovery specialist, Sally changes the name of the performer for this new task to Charlie. Charlie receives a message in his electronic in-box and looks at the claim using the same windows used earlier by Lucy, Sally, and Linus.

Sally adds a task to Check with Charlie in ten days to remind her to follow up with Charlie.

The next step is to divide the story into logical scenario steps. Use the whiteboard to divide the story into scenario steps. Start with a blank scenario and document each scenario step using the “Step Description” column. For the insurance claim example, the whiteboard might look like this when the group finishes (of course, you will not have the time to be as verbose as this example

#	Step Name	Actor	Triggers (Events and rules)	Data Inputs (Information)	Step Description	Data Outputs (Information)	Metrics (Performance, etc.)
10					<ul style="list-style-type: none"> <li>- Lucy, a call center agent, receives a phone call from a Pizza Place employee to notify ACME Insurance of a basic workers' compensation claim for a cut finger.</li> <li>- Lucy sets up a new claim and enters the name of the insured, the policy number, the state in which the incident occurred, and the date of loss.</li> <li>- Lucy saves the claim. The system validates the claim information...</li> </ul>		
20					<ul style="list-style-type: none"> <li>- ... and automatically assigns the claim to Linus (as the owner) because he is the claim supervisor for the state in which the incident occurred.</li> </ul>		
30							
40							

It is a good idea to complete the remaining columns for at least two or three scenario steps. It gives you a chance to discuss each column and its importance. Emphasize to the group that they will be using this information later ( e.g., to identify procedures, business policies, etc.). For the insurance claim example, the whiteboard might look like this when the group finishes:

#	Step Name	Actor	Triggers (Events and rules)	Data Inputs (Information)	Step Description	Data Outputs (Information)	Metrics (Performance, etc.)
10	Receive notice of claim	- Claim call center handler	- Phone call from Pizza Place employee	- Employee Number - Description of injury - Account information	- Lucy, a call center agent, receives a phone call from a Pizza Place employee to notify ACME Insurance of a basic workers' compensation claim for a cut finger. - Lucy sets up a new claim and enters the name of the insured, the policy number, the state in which the incident occurred, and the date of loss. - Lucy saves the claim. The system validates the claim information...	- A new claim	- Length of call
20	Assign claim to owner	- Computer system	- New claim without owner	- Claim - Assignment rules - Potential owners	- ... and automatically assigns the claim to Linus (as the owner) because he is the claim supervisor for the state in which the incident occurred.	- Claim assigned to owner	

You may decide to introduce scenarios using an example from outside the problem domain (e.g., eating breakfast). Sometimes it is easier to get your participants involved.

## 5. Document the remaining scenarios in small teams.

Break into small teams. It is important for each team to have a business expert in the area being discussed. Assign two or three scenarios to each team. If you have not completed the baseline scenario, assign it to one team.

Because it is unrealistic to gather enough scenarios to completely cover the business process, the most common concern regarding scenarios is the fear of missing requirements. At the end of the session, if your participants are still worried about missing requirements, ask them to mark a scenario step with a “V” to signify that more variability exists within this step.

Another common question is whether or not each scenario must begin at the same place and end at the same place. For example, for the Fulfill Order business process, must each scenario begin when the order entry clerk enters an order and end when the order is shipped? The answer is “No.” Otherwise scenarios would tend to be overly redundant. It is acceptable to document “mini” scenarios for parts of the business process with a lot of variability. This approach could lead you to one baseline scenario, two or three other scenarios that span the business process, and four or five “mini” scenarios that address pockets of variability.

**6. Wrap up session.**

Document the scenarios.

### 5.7.3 Sample Workflow Variability Matrix

Scenario Category [1]	Scenario [2]	Sources of Variability [3]										
		Customer	- Post-bill	- Pre-bill	- Preferred	- Not Preferred	Product	- No substitute	- Substitute	Inventory	- Primary w/h	- Alternate w/h
			[4]									

#### Sample Description

**[1] Scenario Category**---Type of scenario referenced.

**[2] Scenario**---Name of the workflow scenario (may be a link in BI Designer).

**[3] Sources of Variability**---A factor or condition that causes a change in the way a business process is executed. You may also decide to use variability categories to group sources of variability into categories.

**[4] Checkmark**---Indicates that a particular scenario will address a particular source of variability

## **5.8 Workflow Validation Guidelines**

### **Description**

This guideline provides guidelines on a generic process for validating workflow diagrams and guidance to the techniques that may be applicable.

### **When Applicable**

Use this guideline when workflow design has been completed and the workflow diagrams are ready for validation.

### **Tools**

Process Simulation Tools

### **Content**

Workflow validation is necessary to:

- Ensure the right tasks are identified in the workflow
- Ensure there are no gaps in the workflow
- Ensure the process can perform as expected
- Check the usability of the workflows
- To obtain buy-in top to the workflow designs

As a guideline, workflow designs should be tested when they are 70% complete, meaning when the design team is largely satisfied that the remainder of the design would be completed either:

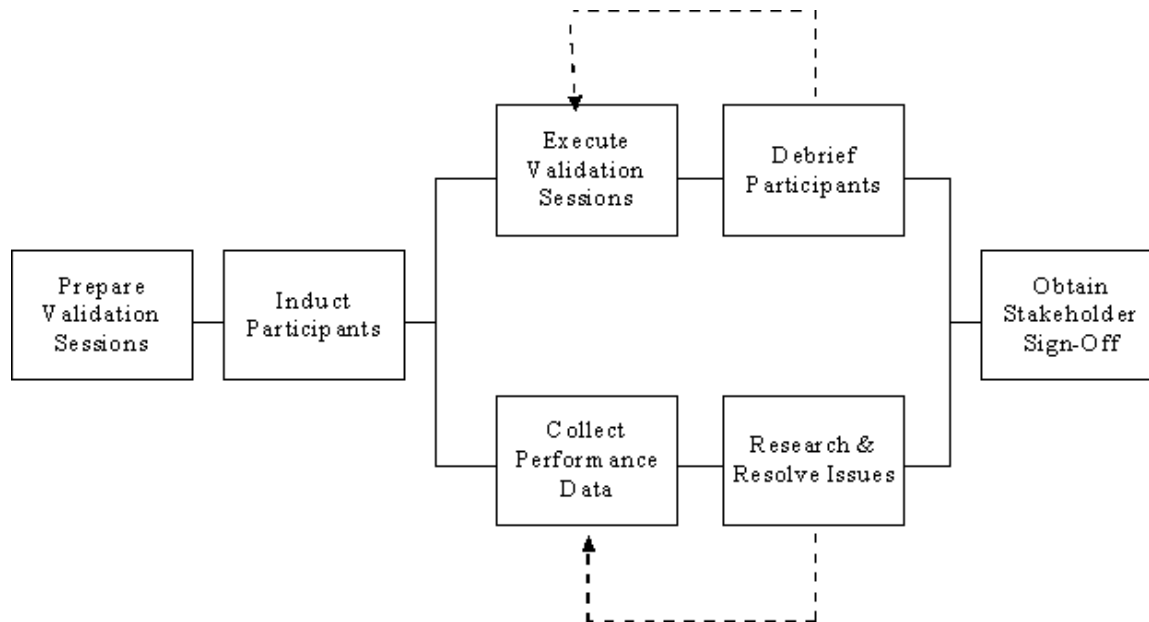
- As a result of the validation session
- During implementation

There are various ways of validating workflows.

This guideline includes:

- A generic process for validating workflow diagrams
- Description of techniques to use when validating workflow diagrams

### **Workflow Validation: A generic process**



### Prepare Validation Sessions

A central coordinator should schedule the Validation Sessions in advance in order to ensure the attendance of participants. This will also avoid double-booking as often more than one process may be tested concurrently. Ensure the right people are involved in the validation. Generally, choose people who will have a role in the future organization, people who have not been involved in the workflow design and people from the various areas of the Organization. Ensure logistics considerations are accounted for and key users have been identified and reserved for the validation sessions.

During preparation of the validation sessions, prepare the validation inputs. The validation inputs will depend on the validation technique chosen (see Techniques for Workflow Validation section of this document).

### Induct Participants

Participants should be provided with materials for the workflow validation prior to the session (usually 2-3 days in advance). Execute an introduction to the session prior to the actual validation and cover the following topics:

- Objectives of the validation session
- Overview of the business process design
- Overview of the validation inputs (scenarios, workflows, etc.)
- Overview of the validation process
- Role of the participants
- Role of the facilitators



### Execute Validation Sessions

Execute validation using the chosen technique. Do not allow the workflow designers to aid the participants in the validation. Use a facilitator to guide participants, ensuring they focus on the task at hand rather than trying to redesign the process. Log any issues that arise as part of the validation session. Assign the logged issues to individuals for resolution after the session.

During the session, focus on:

- How well participants perform the tasks
- Interfaces between processes, activities, and tasks

### Collect Performance Data

Collect performance data based on the expected results developed in the validation approach and the business process metrics that have been defined. Consider:

- What is the average total time required to transform an input/trigger into a final outcome/output?
- What are the costs of the resources that are used for each task in the Workflow?
- What is the average unit cost of completing each task?
- How often are resources left unused/idle?
- How busy are the resources/how often are they in use
- How many units of resources are required to complete each task within a given time frame?

The results should be evaluated against the following criteria:

- Achievement of the Business Capability Requirements for Process
- Achievement of targets defined in the Business Performance Model

Dependent upon the validation techniques, the validation results will vary in nature. For example, using a software simulation, the results may be statistical in nature, but with a conference room pilot, they may take the form of a confirmation/sign off that Business Capability Requirements for processes have been met.

Example criteria that can be used to validate the Workflow Diagrams are:

- Non-Value Added Activity has been removed
- Accountabilities have been outlined by Process Owner
- Design is Process driven, not driven by task or functional area
- Management processes reflect the Process Owner role
- Workflows enable multi-competency, cross-functional teams
- Performance measures are accurate, outcome focused and encourage the right behavior
- Tasks have been combined and handovers reduced

### Debrief Participants

Following the validation sessions, debrief participants based on the issue log developed throughout execution of the validation. Resolve any issues which can be addressed immediately. Update workflow designs to reflect resolutions.

### Research and Resolve Issues

Where an issue cannot be resolved through the debrief session, enlist the required help to resolve the issue and update the workflow diagram designs.

### Obtain Stakeholder Sign-Off

Obtain formal sign-off before developing the workflows into Capability Interaction Models.

## **Techniques for Workflow Validation**

Techniques for workflow validation include scenario based validation, electronic business process simulation, or consensus debrief (workshops).

Use the following definitions to gain a high-level understanding of workflow validation techniques.

### **Scenario Based Testing**

Scenario testing simulates “live” working conditions of the new workflows. Based on research, “real” scenarios are created which represent a realistic series of events and are then executed by participants. The scenarios should represent the full scope of the business processes, including all of their variabilities. The scenarios test the workflows, the tasks within the workflows and the interface of various workflows. Expected results are documented and compared to actual outcomes after the validation sessions.

To understand more about creating scenarios see guidelines: Scenario Gathering Technique, Scenario Planning Technique, and Using a Variability Matrix to Document Workflow Scenarios.

### **Process Simulation**

Process Simulation tests the performance of the process under simulated real-world conditions. It takes a workflow perspective and prototypes a piece of, or the entire, process at any level of desired decomposition in order to determine if the integration of the various resources with the workflow will operate as intended. This technique can be used to predict throughput, service levels and resource efficiency of the workflows. Validation results will typically be statistical in nature.

### **Consensus Debrief (Workshops)**

Consensus Debrief is a technique that can be used to identify the weaknesses and/or strengths of a design. It involves 10-20 people working in groups. The participants should include key stakeholders in the sponsoring organization, subject matter experts, and designers from dependent areas of work (e.g. usability experts). The objective of the consensus debrief is to obtain a consensus on the final design. Issues raised should be captured and prioritized in order of criticality. Together, the group decides how critical the issues are. The major steps involved in undertaking a consensus debrief are:

#### **Plan:**

- Form a group for each of the Processes and their associated tasks
- Plan the opening question, i.e. does the design meets the business capability requirements
- Identify the facilitator
- Plan and confirm logistics

#### **Conduct:**

- Walk through the Workflow designs
- Generate issues
- Identify strengths and weaknesses
- Clarify Issues
- Consensus Voting
- Rank issues

The output of this process should be a list of strengths and weaknesses of the design. The design can then be iterated until issues are resolved.

## 5.9 Deliverable Matrix

The deliverable matrix shows each deliverable and the task where it is created (C) and updated (U).

Deliverable	Deliverable Description	Collect & Manage Requirements	Document & Categorize Requirements	Prioritize Requirements	Allocate & Baseline	Define New or Modified Business Capability	Define Business Capability Performance	Define Process Flow	Define Scenarios and Variability	Architect and Configure Release	Perform Impact Analysis	Select Packaged Software	Define Custom Components & Data Entities	Identify Modified & Reused Configuration	Validate Release Configuration	Document Release for Vendor Bids	Develop RFP / RFI	Screen and Survey Vendors	Evaluate and Select Vendor
Business Capability Requirements	The requirements developed before, during, and after this stage of the process that form the basis of the release. These will be stored in a requirements management tool.	C	U	U															
Business Capability Validation Code	This deliverable will clearly state the conditions against which the business capability (and business capability			U		C								U					
Business Performance Model	The performance specifications of the business capability when delivered. It includes performance measures and					C													
Business Process Workflow	The business practices flow of work for the new business capability.							C											
Dialog Flow Diagram	The system interaction flow for the new business capability.							C											
Variability Matrix	The variability matrix describes the transactions and conditions that are handled in the business capability. It is a detailed scope definition for the capability.							C											
Impact Analysis	The list of all of the configuration items which need to be changed to implement the capability.									C									
Requirements Traceability Matrix	Matrix which shows requirements mapped against the configuration items which need to be modified to support the requirements.										C	U							
Software Selection Justification	Document which describes the rationale for selecting specific software.										C								
Entity Relationship diagram	The Entity-Relationship Diagram (ERD) models an application's entities and their interrelationships to provide a basis for further entity analysis, database design, file structure design, and technical design.											C							
Component Configuration Items	The identification, specification, and requirements tracing to new, modified, and reused configuration items is a key control point for SFA. SFA owns the architecture, that is, the structure of the system and SFA should specify new configuration items such as business components or data											C	U						
Business Case	The Business Case provides economic justification for the new/modified business capability. The Business Case explains why the sponsoring organization must change, what value it receives by changing, and what steps are necessary													U					
Request for Proposal (RFP)	This deliverable prepares a request to submit a proposal for providing goods or services to a SFA. The request describes the item(s) being procured and all associated requirements (e.g., quantities, delivery dates, warranties, etc.)																C		
Request for Information (RFI)	This deliverable contains general questions (based on screening criteria) to send to vendors so that background information may be obtained about their company and its packaged software products.																C		
Task Order	Document which contains the task overview, objectives, approach, timeline, and cost.																		C

## 5.10 Entry and Exit Criteria

This section contains detailed Stage Entry and Exit for each stage of the Capability Development process. These should be tailored to match the scope and requirements of each project that uses the criteria.

### 5.10.1 Capability Analysis Stage

#### 5.10.1.1 Entry Criteria

##### Journey Management

Key stakeholders have accepted the Business Case and IRB Proposal, and are ready to proceed with implementing the capability:

- They have formally expressed their acceptance and support for the plans.
- Their acceptance is visible to the organization as appropriate, such that the organization knows that the new capability and program has the support of the key stakeholders.
- The stakeholders agree with the performance measures and targets expressed in the Business Case.

The appropriate level of leadership and ownership exists to embark on the Capability Development process:

- The change journey sponsors have identified the next level of leadership in the change journey; these are the personnel who will have line operating responsibilities for the new business capability.
- The change journey sponsors have identified the next level advocates and sponsors in the change journey; these include the people who will operate the new business capability.
- The work plan and resources required to implement the next levels of the change network are in place and functioning.

Executive leadership and the sponsor have provided the required amount of organizational resources to the Capability Development process:

- They have identified the personnel, by name, who will participate in the early stages of the Capability Development process, and have committed to the overall resource levels as defined by the IRB Proposal.
- These targeted personnel know that they will be working on the Capability Development process.
- Access to sponsors, workforce, subject matter experts, customers, suppliers, and other information resources is secured.
- All resources have attended orientation sessions and are acquainted with the program and business capability.

The PMO has an effective means of defining and assessing the benefits and costs contained in the Business Case:

- The sponsors and executive leadership have accepted the Business Case.
- Business Case assumptions are clearly documented and communicated across the program.
- There is a defined and approved process for managing changes to the Business Case.

##### Program Management

The Capability Analysis work plan is complete and accepted by the program sponsors:

- The program's costs are estimated and deemed within the tolerance of the Business Case.
- The projects within the program for Capability Analysis are identified, estimated, scheduled, and resources have been staffed.
- Approximations for the projects and resource requirements for the remaining stages are complete and within the tolerance of the Business Case.

##### Business Capability

The Operating Strategy of the business capability owner provides a clear direction that will be applied to make design decisions for all business capabilities within the business area.

#### 5.10.1.2 Exit Criteria

The estimates and approximations are for the release of this business capability and future releases of this business capability. If concurrent business capabilities are being delivered, then the other business capabilities require separate estimating, under the direction of their corresponding program(s).

Plans have been modified for any quick wins efforts that have been identified since the last revision of the program master plan.

Costs for procuring any facilities, equipment, or other capital items that are part of this or subsequent releases of the business capability have been considered.

Costs (payroll and non-payroll) associated with deployment and training activities, both for the deployment team and the affected workforce have been considered.

Delivery planning effort with all the key stakeholders had been coordinated. For example, ensure IT operations and other key stakeholder interfaces with whom the team can coordinate throughout the rest of the Business Capability development stages have been assigned.

The Capability Analysis stage has been completed successfully before proceeding.

The business capability requirements, process flow, configuration, and variability matrix is internally consistent:

- The required changes implied by the business capability are sufficient to cover the gaps identified in the business case and IRB proposal. For example, the solution raises performance to a degree that is enough for the capability to meet its Business Performance Model.
- The competencies, roles, equipment, and applications fully support the business processes.
- The technical infrastructure fully supports the applications as well as the skill building applications such as computer-based training and knowledge management.
- The performance measures within the Business Performance Model can be readily allocated to human performance, business process performance, application and technology performance measures.
- All interfaces to legacy elements as well as legacy applications are identified.
- The delivering team clearly understands which elements are specific to the business capability and which are reused across the enterprise. For example, a scheduling tool can be used to support a manufacturing capability, or it can be applied on a broader basis to multiple capabilities within the enterprise.
- All dependencies between this capability and others are defined. This means that the capability design is aware of the functionality provided by the infrastructure, as well as the demands that other capabilities will make on this design.

There is an identifiable Business Case related to the business capability:

- The benefits stream from this capability can be identified either in quantitative or qualitative means.
- The Business Performance Model related to the capability reflects the Business Case.
- The Business Case supports the decision to deliver the capability.

The business capability provides a sufficient basis from which to estimate the Capability Design stage:

- All business processes are identified for the business capability.
- All technical architecture which the capability will use are identified.
- All applications required for the capability are defined, and the software engineering approach has been selected (for example packaged, custom, object, or a hybrid).
- The work groups, expressed in terms of the organizational groups with similar types of responsibilities, are identified.

Quick wins related to this business capability are incorporated into the program plan:

- Quick wins identified in the Business Architecture stage are estimated, staffed, and underway.
- Additional quick wins are identified.
- The dependencies between each quick win and the business capability are defined.

The interfaces between the quick win, legacy, and future business capability are defined.

## ***5.10.2 Capability Release Design Stage***

### **5.10.2.1 Entry Criteria**

#### **Journey Management**

The required sponsors, executives, and other stakeholders have authorized the Capability Release Design stage effort, and have:

- Approved any revisions to the release schedule and scope of individual releases
- Agreed to the capability deployment approach and plan
- Agreed to any changes in scope for the business capability
- Authorized the Capability Release Design stage budget and schedule

The change network in support of the capability is operational and in the process of building sponsorship and ownership:

- The network is building ownership among the personnel with line responsibility for the capability.
- The network is building advocates and sponsors among the personnel responsible for operating the capability.

The communications content and plan reflects the decisions and structures made during the Capability Analysis stage:

- The communications reflect any changes to the release structure or timings.
- The communications reflect the results of the implementation decisions made during the Capability Analysis stage.

### **Program Management**

The capability release scope is defined in sufficient detail to proceed with design:

- There are no outstanding implementation issues that would affect the release design. These should have been resolved during the Capability Analysis stage.
- Packaged software has been identified and, in most cases, selected. The implementation effort has been included in the work plan.
- The Business Performance Model deliverable provides a set of measures and targets that are at a level of detail such that we can evaluate the design's potential business performance.
- The capability release scope is feasible given the operating, business architecture, and other constraints.

The team organization structure and work practices are in place and understood by all release design teams:

- Teaming agreements are in place for all distributed development sites.
- The management processes are in place and communicated to all release design teams.
- Orientation materials, education, and processes are in place and operational.

Capability release teams are fully supported by the team work environment:

- The design tools are installed, customized, and operational. This includes tools that support application analysis within this stage.
- The capability release teams have adequate facilities for individual work, as well as meetings and conferences.
- If a model office or business simulation lab is required, it has been built and is operational.
- Resources continue to be staffed in support of the team work environment.

The projects within the Capability Release Design stage are complete and ready to begin:

- The design project work plans are complete and approved.
- The design project estimates are complete.
- The design projects are staffed.
- The relationships and dependencies among the design projects are known and reflected in the project schedules.
- The scope of the individual design projects covers the scope of the capability release.

### **Business Architecture Blueprint**

The Capability Requirements remain consistent with the business:

- The requirements for the capability do not conflict with the operating strategic direction.
- The Capability Release Design stage teams are aware of the strategic direction of the organization.

- The Capability Release Design stage teams are aware of the enterprise-wide implications of the capability release.

The Business Capability Requirements deliverable recognizes and reflects the business architecture constraints:

- The impact of the capability on customers is understood and acceptable.
- The impact of the capability on the enterprise-wide architectures (i.e., organization, competency, delivery vehicle, facilities, and equipment) is understood and acceptable.

### **Business Capability**

The Business Capability Requirements deliverable defines the release in sufficient detail:

- All open issues regarding the Business Capability Requirements are either cleared or identified and communicated to the release design teams.
- The Business Capability Requirements deliverable reflects all approved change requests.
- The Business Capability Requirements cover the solution components identified for this release.
- The Business Capability Requirements reflect specific customizations required for each type of operating location and/or deployment unit.
- The deployment related requirements are sufficient to support deployment and operation of the release.
- The Business Capability Requirements are sufficient to validate the selection of packaged software.

The cross-release dependencies and implications of the current release are identified and deemed acceptable:

- The prerequisites for the release deployment are in place prior to completion of the release.
- The external dependencies for the release are identified and incorporated into the Program Plans.
- The internal dependencies for the release are identified and incorporated into the Program Plans.
- The interfaces and interactions between this release, existing capabilities, and legacy elements (e.g., applications, operating units, organizations) are identified and included within the scope of the Capability Release Design stage projects.

The Capability Analysis stage deliverables provide sufficient information to estimate the Capability Release Design stage projects:

- Business processes within the scope of the release are named and defined.
- Automated applications involved in this release are named and defined.
- Potential sources of content are identified and/or the effort of developing the new content is understood.
- Interfaces to legacy applications and data are named and identified.

- Types of equipment required for the release are identified.
- Competencies for the release are identified and defined.
- Job types are identified and defined.
- Corporate data model provides a means for estimating the number of data entity types.
- The number and type of delivery vehicles are defined.
- Packaged software vendors have been identified and contracts have been negotiated.
- The number and distribution of operating locations (including central sites) is known.
- The Technical Architecture deliverable identifies the number of execution environments and Technology Infrastructure tiers.
- When using a component-based development approach, the business components in-scope have been identified.
- The Organization, Competency, and Culture requirements are sufficient to begin defining the human performance infrastructure.

For cases where the capability streamlines business processes:

- Current business processes were analyzed during the Capability Analysis stage to identify opportunities for streamlining.
- Current business processes were analyzed in sufficient detail to benchmark their performance for comparison against the new processes.
- Requirements reflect the opportunities to apply application technology, equipment, and role responsibilities to streamline the business process.
- The Business Performance Model deliverable reflects the level of detail necessary for the teams to design to meet the streamlining requirements (for example they express specific process performance times, resource requirements etc.).

#### **Other**

Operational aspects of the business capability are incorporated into the Capability Release Design stage projects:

- The Business Case reflects operating costs associated with the capability.

#### **5.10.2.2 Exit Criteria**

If applicable, the standard work plan templates and estimating models have been updated to take into account the experience gained in any previous releases or any concurrent release work that may have occurred.

It may be necessary to receive authorization for all work during the Capability Release Build and Test stage, including the ongoing program-level activities that are not aligned with the Delivering stage. If so, it will be necessary to identify and

baseline the set of work to be performed in conjunction with the Capability Release Build and Test stage.

The estimates and approximations are for the release of this business capability and future releases of this business capability. If concurrent business capabilities are being delivered, then the other business capabilities require separate estimating, under the direction of their corresponding program(s).

Quick wins efforts that have been identified since the last revision of the program master plan are planned for.

Costs for procuring any facilities, equipment or other capital items that are part of this or subsequent releases of the business capability have been considered.

Costs (payroll and non-payroll) associated with deployment and training activities, both for the deployment team and the affected workforce have been considered.

The Capability Release design supports the realization of the Business Case and Business Performance Model.

Formal documentation from key sponsors and stakeholders is obtained signing off on the work completed in the Capability Release Design stage and authorizing the start of work in the Capability Release Build and Test stage.

The Business Case has been updated to reflect the revised costs, release scope and release milestones, as necessary.

The business capability design is baselined; that is, after this point, all design changes are governed by the configuration management (i.e., change request) process

### ***5.10.3 Capability Release Build and Test Stage***

#### **5.10.3.1 Entry Criteria**

#### **Critical Questions**

There are certain things that must have occurred in the Capability Release Design stage in order for Capability Release Build and Test projects to commence. Answer the following critical questions to determine whether the Capability Release Build and Test stage should begin:

Have all Capability Release Build and Test teams been identified, including candidates for the roles within them? Ideally, individuals who have worked on the Capability Analysis and Capability Release Design stages should work on Capability Release Build and Test teams. This is to ensure continuity throughout the development of the capability.



Have the Capability Release Build and Test work effort, completion schedules, and estimates been approved?

Has the Capability Release Build and Test work environment been established?

Is the technology infrastructure in place?

Are all necessary test plans, test conditions, and expected results in place?

Have the standards that the Capability Release Build and Test stage will adhere been agreed upon and published?

### **Outputs That Must Be Available**

The following outputs must be available in some form and must be complete and available for the Capability Release Build and Test projects. For samples refer to Method/1:

- Application Architecture
- Application Requirements Specification
- Business Capability Requirements
- Business Performance Model
- Business Process Design
- Capability Interaction Model
- Data Conversion Design
- Database Design
- Content Model
- Media Content Design
- Organization Infrastructure
- Packaged Software Design
- Packaged Software Evaluation
- Physical Environment Designs
- Performance Enhancement Infrastructure
- Technology Infrastructure Design
- Technology Infrastructure Component Requirements
- User Interface Design
- User Interaction Model
- Work Unit Inventory

### **Outcomes That Must Have Occurred**

The following outcomes associated with a Business Capability Release and the projects contributing toward it are not essential for the Capability Release Build and Test projects to commence. However, it is highly desirable that they are achieved. The outcomes are:

Sponsoring organization commitment to the result

Quick wins implemented and demonstrably adding value to the sponsoring organization

Sponsoring organization involvement in the design process

Sponsoring organization commitment to build

### **Risk Assessment**

Any deliverable failing to meet its criteria is returned to the stage that developed it.

Program and project management should be fully aware of any circumstances in which Capability Release Build and Test Stage Entry Criteria have not been met, yet the Capability Release Build and Test activities will still be taking place.

### **5.10.3.2 Exit Criteria**

#### **Business Capability Costs and Benefits**

The business capability, as defined in the Business Case and in the Business Capability Requirements, can be achieved. It meets all stated user requirements. In addition to meeting requirements, the benefits from the capability will be achieved.

Benefits and costs associated with deployment are acceptable and in line with the Business Case. In measuring the actual benefits, keep in mind that the full benefits might not be realized initially due to learning curve effects.

The team has confirmed the costs associated with deployment of the capability release.

The team has established metrics to monitor the capability release upon deployment.

#### **Stakeholder and Sponsoring Organization Acceptance**

The stakeholders have sufficient confidence in the capability release to decide to deploy. Their expectations of the business capability are met. In particular, the internal and external stakeholders of the capability, as well as executive and operational staff, have bought in to the capability release and have established ownership of it.

Operating units in different countries with different cultures and languages accept the capability release.

The sponsoring organization accepts the human implications of the capability release.

The key stakeholders accept the new processes and working practices and agree that deployment of the capability is the next step to take.

### **Deployment Prerequisites**

The team has completed training materials for the deployment and data conversion teams.

The deployment approach has been validated.

Business continuity has been guaranteed (i.e., existing processes will be allowed to continue as normal).

The risks associated with deployment of the capability release are mitigated appropriately.

The deployment procedures and software can install the new capability in the production environment.

The capability release can be operated in the production environment.

The team has discovered and addressed significant maintainability and operability issues associated with operation of the capability release.

The team has resolved all significant business integration issues associated with deployment of the capability release.

The team has documented all lessons learned during verification and validation that affect deployment.

## ***5.10.4 Capability Deployment Stage***

### **5.10.4.1 Entry Criteria**

#### **Management Sponsorship**

Has sponsoring organization commitment to the deployment been obtained?

Is management sponsorship to deploy the new business capability visible?

Is there buy-in from the sponsoring organization? This involves buy-in from senior management, line management, and staff.

#### **Program and Project Management**

Are the deployment approach and deployment plan ready?

Is the definition of a deployment unit available?

Have necessary checkpoints been identified to support the deployment of the business capability release?

Have the risks associated with deployment of the business capability release been identified and analyzed? Has a mitigation plan been developed for them?

Is a contingency plan available in case the business capability release is not deployed as planned?

Have expectations of the business capability release been clearly and correctly conveyed to the sponsoring organization?

Has a deployment team been created? Is the size of the deployment team consistent with the size of the deployment unit?

Is a schedule and plan for deployment available?

Is a change control/configuration management mechanism available? Has the deployment team been trained to use it?

### **Overall**

Is each deployment unit aware of the deployment approach and deployment plan?

Are the results of the deployment assessment available?

Is there a mechanism to capture experiences from the current deployment unit and feed it back to management?

Are the pilot implementation experiences documented and available for deployment?

Is the deployment package tested and ready? Check for quantity, completeness, and consistency. Check that the deployment package contains:

- All elements of the application
- All elements of the business policies and procedures
- All elements of the physical environment
- All elements of the technology infrastructure
- All elements to transition the workforce

Have all elements of the deployment package been tested? Do they meet the required quality levels?

Is a verification plan to test the deployment of the business capability release to the deployment unit available?

Have all the key performance indicators of the business capability release been documented?

Have the critical success factors for deployment of the business capability release been documented and been signed off by the sponsoring organization?

### **5.10.4.2 Exit Criteria**

The business capability elements are properly installed and ready to be activated.

The deployment team is prepared to “flip the switch”.

Contingency planning is understood, and a clear communication mechanism is in place for indicating when to use the contingency plan.

The consolidation and analysis of performance measurements is part of an iterative process to eliminate irregularities and allow for the resolution of issues.

If applicable, the legacy systems is removed from “old” operations and management processes after having made the irreversible decision to proceed with the new business capability. If performed, it is critical to estimate this task accurately, as it involves the coordination of internal and external resources and execution of backup and archiving policies.

The monitoring of planned progress has verified the stability of the deployment unit in using the deployed business capability release, and ensured that a “ready to release” state has been reached. This state is verified by analyzing the actual performance and productivity forecasts of the deployment unit. The verification of stability should not be viewed as a formal benefits realization test that is performed against the Business Performance Model, but rather, as a verification of a consistent rate of change toward the planned benefits.